BIBLIOTECA BASICA INFORMATICA

MS/DOS



el estándar de IBM

INGELEK

BIBLIOTECA BASICA INFORMATICA

MS/DOS

12 el estándar de IBM

INGELEK

Director editor:
Antonio M. Ferrer Abelló.

Director de producción:
Vicente Robles.

Coordinador y supervisión técnica:
Enrique Monsalve.

Colaboradores:
Angel Segado.
Patricia Mordini.
Margarita Caffaratto.
Marina Caffaratto.
Francisco Ruiz.
Jorge Juan Monsalve.
Beatriz Tercero.
Fernando Ruiz.
Casimiro Zaragoza.

Diseño:
Bravo/Lofish.

Dibujos:
José Ochoa.

ISBN del tomo: 84-85831-47-0
ISBN de la obra: 84-85831-31-4
Fotocomposición: Pérez Díaz, S. A.
Imprime: Héroes, S. A.
Depósito Legal: M-1806-1986

INDICE

PROLOGO

Cuando nos hemos sentado ante nuestro ordenador personal IBM y nos hemos preguntado cómo orientar lo que teníamos que escribir (con el tratamiento de textos) nos hemos dado cuenta del amplísimo margen en el que nos podíamos mover. Sin embargo, resulta imposible, por falta de espacio, explicar todos los términos que vamos a utilizar; por ello hemos partido del supuesto de que los lectores de esta monografía tienen ya un conocimiento básico mínimo que les permite comprender palabras tales como "fichero, registro, programa, formulario, microprocesador", etc. Quienes hayan leído las monografías anteriores de la BBI no deberían tener problema alguno a tal respecto. A pesar de ello hemos tratado en todo momento de ser claros y sencillos en nuestra exposición.

Por qué un sistema operativo

Desde siempre el hombre ha soñado con poder disponer de un esclavo dócil, servicial, siempre obediente y que nunca se cansará de satisfacer sus deseos y necesidades; quizá con la era de los ordenadores haya logrado, casi en su totalidad, hacer realidad dicho sueño. En nuestros días estamos rodeados por muchos "sirvientes" que alivian nuestras fatigas y entretienen nuestro ocio: resulta casi imposible renunciar en una casa a la lavadora, la batidora, el aspirador, el frigorífico, etc., como también es imposible renunciar en la oficina a todos los instrumentos que la civilización nos ha proporcionado: télex, teléfono, máquina de escribir, fotoco-

piadora y, ahora, el ordenador. Todos sabemos que existe el ordenador, pero no todos sabemos qué puede hacer y cómo es posible utilizarlo con el máximo beneficio.

El ordenador lo podemos considerar como un esclavo que, a petición nuestra, nos proporciona todas (o casi todas) las informaciones que queremos, nos prepara atractivos gráficos, nos ofrece las estadísticas de las que tenemos necesidad cuando las solicita nuestro jefe, nos permite enviar un télex aunque la secretaria se haya ido a casa, y otras muchas prestaciones, pero...

Su único defecto es que comprende las cosas al pie de la letra (alguien lo ha definido como un perfecto idiota). Desde luego, es un ejemplo de total obediencia, pero precisamente por esto tenemos que darle órdenes con tal formalismo y precisión que, en ocasiones, nos hace perder la paciencia.

Los primeros ordenadores se programaban en los más mínimos detalles y todas las instrucciones tenían que escribirse una tras otra con perfecta lógica; de no ser así, los resultados eran nefastos. Esto quería decir que cada vez era más enojosa y compleja la escritura de instrucciones (incluso actualmente hay pequeños ordenadores que para funcionar deben ser programados de este modo). A medida que los sistemas se hacían más potentes y sofisticados, la tarea de programar era cada vez más difícil y, por consiguiente, exigía más tiempo y se hacía cada vez más costosa. Era un auténtico círculo vicioso.

El remedio, afortunadamente, no se hizo esperar, y fue así como comenzaron a hacer su aparición los primeros sistemas operativos. En definitiva, no son otra cosa que nuevos esclavos (aunque esta vez formados mediante el software) a los que se les confían las tareas más enojosas y difíciles.

No obstante, tienen un inconveniente, y es que exigen mucha memoria. Este es uno de los motivos por el que los pequeños ordenadores, dotados de poca capacidad de memoria, tienen muchas veces solamente unos rudimentarios y muy básicos sistemas operativos.

Uno de los cometidos más importantes que se le confían a un S.O. es encargarse de toda la problemática del control de las unidades periféricas (sobre todo de las de disco); así ya no es necesario especificar "con pelos y señales" lo que el ordenador debe hacer, porque es el sistema operativo quien interpreta nuestras órdenes y procede luego a proporcionar al microprocesador todas las instrucciones, una a una. Nosotros decimos: escribe, lee, abre el fichero..., y el sistema operativo procederá dócilmente a ejecutar, en sus más mínimos detalles, estas operaciones. A veces será tan gentil que nos avisará que hemos olvidado algún parámetro y de forma educada nos pedirá que pongamos remedio a tal situación.

Es el sistema operativo quien se encarga de gestionar adecuadamente el espacio de memoria interna, de enviar los textos a la impresora, de leer o escribir información en la memoria de discos, de reconocer las diversas órdenes que proceden del teclado, de enviar y visualizar informaciones en la pantalla y de otras numerosas funciones que tienen como finalidad optimizar la explotación de los recursos internos y externos del ordenador.

Pero, ¿no será este pequeño monstruo demasiado entrometido? No, porque es también muy discreto. Por ejemplo, cuando como usuarios queremos ejecutar un programa es él quien procede a cargarlo en memoria y, después de haberle cedido el control, se retira de la forma más discreta. Desde este momento, el control de la memoria y de todos los recursos disponibles ha pasado al programa que se cargó. Al final de la ejecución, o si lo llamamos, el sistema operativo vuelve a encargarse del control de la situación.

Hasta aquí hemos tratado solamente de las funciones elementales, pero hay otras más sofisticadas que pueden solicitarse por medio de palabras-clave sencillas.

Antes de emprender el estudio de uno de los más difundidos sistemas operativos para ordenadores personales, queremos recordar algunas nociones fundamentales. Los lectores que ya las poseen (acaso por haber leído las monografías anteriores de nuestra biblioteca, incluyendo el volumen 10, que efectúa una visión panorámica de los sistemas operativos) pueden saltar directamente al comienzo del primer capítulo.

En un ordenador personal profesional encontramos los componentes siguientes:

Hardware

- Unidad central de proceso (CPU-Central Processing Unit),
- teclado,
- pantalla de presentación visual CRT (Cathode Ray Tube-Tubo de rayos catódicos),
- impresoras (una o varias),
- memoria de masa de cinta magnética (una o varias),
- memoria de masa de discos (una o varias).

Todos estos componentes pueden combinarse entre sí en el modo que el diseñador haya pensado que es más funcional, atractivo, económico o ergonómico, de conformidad con lo dictado por la moderna Ergonomía. Esta ciencia determina las características físicas a cumplir en la fabricación de máquinas, de modo que se adapten lo más posible a las exigencias del hombre, tanto desde

el punto de vista físico como desde el punto de vista mental. Lamentablemente, suele suceder todavía en nuestros días que ha de ser el hombre quien se adapte a la máquina.

Software

- El Sistema Operativo propiamente dicho,
- programas de servicio o de utilidades,
- programas de aplicación.

De los programas de servicio hablaremos más adelante (en realidad, se suelen suministrar en el mismo disco que contiene el sistema operativo). De los programas de aplicación decir, de momento, que son paquetes de software que nos permiten obtener los datos y programas que nos interesan: nóminas, facturas, estadísticas, gráficos y, por qué no, juegos de ordenador más o menos divertidos e inteligentes. Y ello es así porque también en la vida con el ordenador podemos mezclar el juego con el trabajo.

CAPITULO I

SISTEMA OPERATIVO DE DISCO (DOS): OTRO ESCLAVO DE LA ERA ELECTRÓNICA

La cuna del MS-DOS (o PC-DOS)

Acabamos de ver cómo y por qué nacieron los sistemas operativos; veamos ahora cuándo nació el sistema operativo MS-DOS, que es el estándar adoptado por IBM para su IBM-PC. Estos pequeños antecedentes históricos no nos vendrán mal, aunque no tienen por qué preocuparse, ya que no nos remontaremos a las guerras púnicas, sino que nos limitaremos a una serie de informaciones sobre cuándo y por qué nació el DOS (Disk Operating System, sistema operativo de disco).

Un día no muy lejano IBM (y no creemos que haya necesidad de presentaciones) decidió entrar en el mercado de los ordenadores personales y eligió un microprocesador de 16 bits en lugar de los habituales de 8 bits utilizados hasta ese momento.

Se halló así en la necesidad de encargar o de producir un sistema operativo que controlara el microprocesador Intel 8088 que había elegido para su nueva criatura. Esta fue bautizada con el nombre de Personal Computer (PC para los amigos).

IBM decidió proporcionar al PC tres sistemas operativos: el tradicional CP/M-86 de Digital Research, el UCSD p-System y un nuevo sistema operativo, que llamarían PC-DOS, encargado a Microsoft para su desarrollo.

Es evidente que IBM tomó en consideración, en primer lugar, el tradicional mercado de los programas de aplicación, que se orientaba sobre todo al CP/M-86 (heredero, en el mundo de los 16 bits, del glorioso CP/M-80 nacido en la generación de los 8 bits), pero también se dirigió en una nueva dirección, por un

camino que recorrería con determinación y, ahora ya no puede negarse, con muy buena acogida por parte del usuario.

En el momento de su presentación, el PC hubiera podido utilizar indiferentemente los tres Sistemas Operativos, pero la casualidad o el destino quiso que solamente estuviera preparado el PC-DOS. Esta circunstancia fue determinante para toda la "carrera" del ordenador personal IBM-PC y del mismo sistema PC-DOS.

IBM llamó al sistema operativo que proporcionaba con el PC, PC-DOS, mientras que la misma versión, con diferencias mínimas, se comercializó con el nombre de MS-DOS (MicroSoft DOS); por ello, desde este momento, para no dar lugar a confusiones, utilizaremos el término DOS o MS-DOS tanto para hablar del MS-DOS como del PC-DOS, ya que son "el" DOS sobre el que se centra este libro (hay que tener en cuenta este matiz, pues, evidentemente, sistemas DOS hay muchos —DOS es un nombre genérico—).

Entre estos dos sistemas operativos DOS y el CP/M-86, las diferencias, al menos en las primeras versiones, casi no se podían detectar en la práctica. En efecto, ambos controlan la misma máquina con muchos comandos semejantes. Las únicas diferencias importantes estaban en las prestaciones que un ordenador personal podía suministrar al trabajar con uno o con otro.

La semejanza era consecuencia del hecho de que ambos sistemas operativos tienen como progenitor al CP/M-80 de Digital Research, que, aunque originalmente no era adecuado para utilizar el microprocesador 8086 (de 16 bits), tomó dos caminos diferentes para llegar primero: CP/M-86 y 86-DOS, muy similares.

Los diseñadores pretendían aprovechar lo más posible el software ya disponible para los microprocesadores 8080/Z-80 que, en esa época gozaban de una rica biblioteca. Es precisamente partiendo de esta exigencia común como el CP/M-80 fue reescrito, en el caso del CP/M-86, manteniendo invariable el sistema de gestión de las memorias de masa; a través de este camino se creó el CP/M-86.

En nuestro caso, por el contrario, aunque teniendo presente dicha exigencia, se comenzó por modificar el control de la memoria de masa de discos magnéticos hasta llegar al MS-DOS, que tiene un sistema de control completamente diferente al del CP/M-86. Por sistema de organización de la memoria de masa de discos magnéticos, por si alguien no lo sabe, se entiende el modo en el que están organizados los índices de los ficheros grabados en los discos y la tabla de las asignaciones de espacio para cada uno de estos ficheros.

Este nuevo sistema tiene algunas ventajas, pero también, como casi siempre sucede, algunas desventajas que se ponen de manifiesto en ciertas condiciones operativas especiales. Entre las ventajas podemos, sin duda, incluir un acceso a los datos de un fichero muy rápido, y la constante actualización en el índice de los ficheros del número de los caracteres que el fichero ocupa en el disco y la fecha y hora de la última modificación. Además, se puede pedir al sistema DOS que lea y escriba informaciones (registros) de cualquier longitud.

Entre los inconvenientes ha de citarse la posibilidad de una no del todo correcta utilización del espacio de disco disponible. De ésta desventaja no hablaremos por cuanto se trata de un problema que se plantea solamente en algunas condiciones operativas especiales. Otro inconveniente puede ser el riesgo de destruir irremediablemente el contenido de un disco flexible si, desgraciadamente, alguien poco previsor lo sustituye antes de que el programa de aplicación haya cerrado todos los ficheros. Sobre este riesgo volveremos a insistir, porque la integridad de los datos es "sagrada".

Estructura del MS-DOS: un poco de vocabulario

Veamos ahora cómo está estructurado nuestro sistema DOS (Fig. 1) y cuáles son los componentes de esta estructura:

- IBMBIO.COM: Es un fichero que permite la interconexión de la parte de memoria para sólo lectura (ROM - Read Only Memory) que tiene la denominación de BIOS (Basic Input/Output System - Sistema Básico de Entrada/Salida) y que contiene las denominadas subrutinas de bajo nivel. Se trata de pequeñas subrutinas que permiten la comunicación de los datos entre el microprocesador y todos los dispositivos periféricos del sistema.
- IBMDOS.COM: Es el verdadero programa DOS, que ya no contiene simples subrutinas, sino programas propiamente dichos que permiten abrir y cerrar un fichero, leer y escribir las informaciones procedentes o dirigidas hacia los dispositivos periféricos, bloquear o agrupar (reunir una serie de datos para optimizar el espacio) o desbloquear las informaciones, utilizando los discos y otras funciones.
- REGISTRO INICIAL: Es el primer registro de cada disco ("boot record") e identifica ese disco como disco DOS. Cuando no se encuentra este registro significa que el disco en cuestión no es un disco DOS o bien que es nuevo y no se ha marcado todavía en el *registro inicial* (se emplea el comando FORMAT para efectuar esta operación en todos los discos que hayan de utilizarse en un microprocesador que emplee el DOS).

TABLA VECTOR INTERRUPCIONES
AREA COMUNICACION ROM
AREA COMUNICACION DOS
IMBIO.COM módulo de interface para las subrutinas en ROM de E/S
IBMDOS.COM - subrutinas de gestión de las interrupciones DOS, subrutinas de servicio
Area de tránsito para el DOS, área de control y programas de control de las unidades instaladas
Parte residente del COMMAND.COM, código de recarga de la parte no residente
Comandos externos o programas de utilidad (.COM y .EXE)
Cola de usuario para fichero .COM.
Parte de tránsito del fichero COMMAND.COM. Comandos de tipo interno, intérpretes de los comandos, procesador de lotes y comandos de carga exterior

SEGMENTO MAS BAJO DISPONIBLE

Figura 1.— Mapa de memoria del MS-DOS.

- COMMAND.COM: Es el fichero que contiene la serie de comandos definidos como "internos" (o "residentes") que se hallan permanentemente en memoria. Contiene también el procesador de todos los comandos, incluso los "externos" (situados en el disco, del que se extraen cada vez que son necesarios). Los comandos internos se procesan, lógicamente, con mayor rapidez que los demás, pues estos últimos, antes de ejecutarse, deben ser leídos del disco en donde residen junto con el sistema operativo.
- En el disco DOS hay también programas de aplicación, como por ejemplo el EDLIN (es un programa de escritura, actualización y creación de ficheros línea por línea) y el DEBUG (permite crear, variar, visualizar, ejecutar ficheros en formato, ejecutable —extensión .COM o .EXE— y comprobar estos últimos efectuando variaciones sin necesidad de volver a ensamblar el programa). No debemos olvidar tampoco el intérprete del lenguaje BASIC Microsoft.

El sistema MS-DOS ha sufrido diversas actualizaciones y existen ahora diferentes versiones (1.0, 1.1, 2.0, 2.1, 3.0, 3.1). Nosotros nos ocuparemos fundamentalmente de la versión 2.0, que es una de las más difundidas (sus diferencias con otras versiones más modernas son poco significativas).

En la primera versión (1.0) nada podía hacer pensar en lo que sería el desarrollo del nuevo sistema operativo. Con la exclusión de la diversidad en el control del espacio de disco (lo que, por otra parte, no se conoce si no se comprueban las prestaciones), todo era bastante similar a su competidor directo, el CP/M-86. En la versión 2.0, por el contrario, se comienza a vislumbrar algo nuevo y hay una clara convergencia hacia lo que son las características esenciales del sistema operativo UNIX:

La nueva estructura de los índices (en árbol), una función muy similar a la de "pipe" y la nueva concepción de las unidades periféricas, consideradas como ficheros, son algunas de las novedades de la versión 2.0 que coinciden con las peculiaridades del UNIX.

Otras novedades de la versión 2.0 se pueden resumir como indicamos a continuación:

- Algunos caracteres especiales ya no pueden utilizarse en la definición de los nombres de ficheros porque ahora tienen un significado especial para el sistema DOS (más adelante veremos cuál es este significado). Se trata del signo mayor que ">"el de menor que "<", la barra vertical "|" y la barra invertida "\". Si pretendemos convertir ficheros desde la versión 1.0 a la versión 2.0 será necesario comprobar antes que ninguno de estos caracteres aparecen en sus nombres.
- Aumento de la capacidad de los discos. Es posible grabar unos 20.000 caracteres más en cada una de las caras de los discos.
- Areas de tránsito múltiple (Buffers). El sistema MS-DOS reserva en memoria unas áreas que luego utiliza para almacenar las informaciones que debe leer o escribir en los discos. Estas áreas eran dos en la versión 1.0 y son dos también en la versión 2.0, si no se especifica algo en sentido contrario en el momento de la carga del sistema DOS. No obstante, existe ahora la posibilidad de aumentar a voluntad el número de estas áreas para tratar de optimizar, en cada aplicación, las prestaciones del procesador.

A continuación damos unas características del sistema que son completamente nuevas.

- Fichero de configuración. Es posible crear un fichero que contenga comandos especiales con los que ordenar al sistema que efectúe determinadas operaciones todas las veces que se cargue, tales como, por ejemplo, aumentar el número de las áreas de tránsito (buffer), dar la hora, escribir un mensaje, etc. En este fichero es posible también especificar informaciones tales como los nombres de los programas de control de unidades de expansión u otros parámetros necesarios para el funcionamiento del sistema operativo.
- Unidad de disco duro múltiple. La versión 2.0 introduce la posibilidad de utilizar varias unidades de disco duro (o rígido-hard disk), cada una de las cuales puede subdividirse en particiones. Estas últimas pueden ser usadas por diversos sistemas operativos. En el disco duro es posible cargar el DOS y todos los programas de utilidad. De este modo será posible inicializar el sistema operativo y disponer siempre de todas las funciones y programas sin tener que cambiar los discos.
- Estructura en árbol de los directorios. La posibilidad de utilizar discos rígidos, con notable capacidad de almacenamiento, ha exigido una nueva organización de los índices de los ficheros. En un disco duro se pueden grabar millares de programas; es fácil imaginarse cómo sería el índice si no se encontrara una solución adecuada: simplemente se trataría de una lista interminable de nombres, casi imposible de leer y de interpretar. En la versión 2.0 se puede subdividir el disco duro (y también un disco flexible normal) reagrupando así los ficheros de una misma categoría (Fig. 2).

  Esta solución permite encontrar con mayor facilidad grupos de ficheros, por ejemplo los que queramos utilizar con programas de aplicación tales como Multiplan o Framework, o bien los programas escritos en BASIC, los escritos en FORTRAN, etc.

  Para cada uno de los discos "virtuales" el sistema nos proporciona el índice de los ficheros contenidos en él, o sea, tenemos "subdirectorios" (subíndices).

  Los directorios individuales están separados entre sí como si fueran índices de discos diferentes, pero hay comandos que permiten pasar rápidamente de un índice a otro, leer ficheros pertenecientes a un índice distinto, etc., como si todos los discos estuvieran en línea a su disposición.

  Cada índice puede contener el nombre de otro índice y éste, a su vez, otros. De este modo se crea, a partir del primero, denominado índice-raíz (o directorio-raíz), una especie de diagrama ramificado: de aquí la denominación de estructura en árbol invertido. Organizando adecuadamente estas estructuras resulta posible ordenar al sistema DOS que proceda a leer, escribir o copiar ficheros desde un índice a otro con gran sencillez.

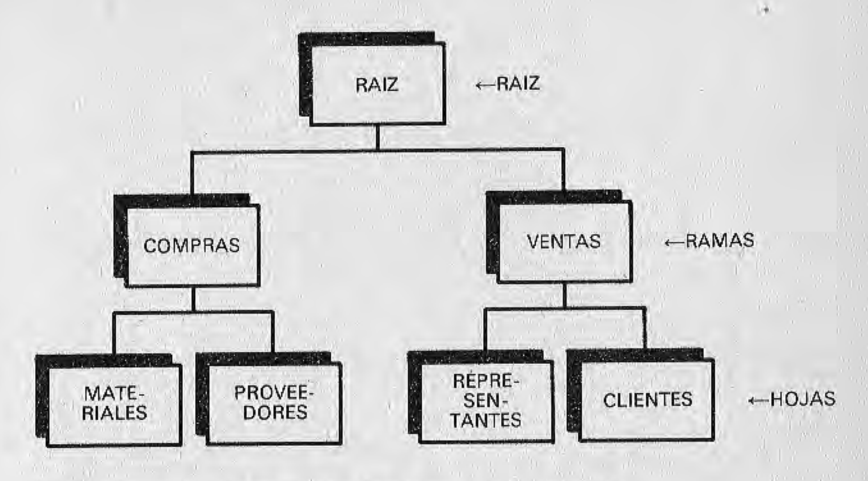


Figura 2.— Estructura simple en árbol invertido.

- Etiqueta para los discos. Cuando se inicializa un disco, es decir, cuando se le prepara para poder utilizarlo con un determinado sistema operativo, es posible etiquetar el propio disco (volumen) con un nombre que tenga como máximo 11 caracteres. Esta etiqueta se pone en el índice o directorio-raíz y es visualizada por diversos comandos.
- Mayor control de la pantalla y del teclado. El MC-DOS permite a los programas de aplicación generar caracteres especiales o secuencias de caracteres que servirán al propio DOS para controlar la función del cursor en la pantalla, la regulación del color y la asignación de significados especiales a teclas arbitrarias del teclado. Por ejemplo, se puede asignar a la tecla F1 una cadena como:

  COPY FICHERO.MIO FICHERO.TUY

  A continuación bastará pulsar la tecla F1 para hacer aparecer en la pantalla el comando completo, evitándonos teclear todos sus caracteres.

• Redireccionamiento de la entrada y de la salida estándares. Para todos los programas que leen datos a partir del teclado y que escriben en la pantalla es posible cambiar estos dispositivos estándar por otros (Fig. 3).

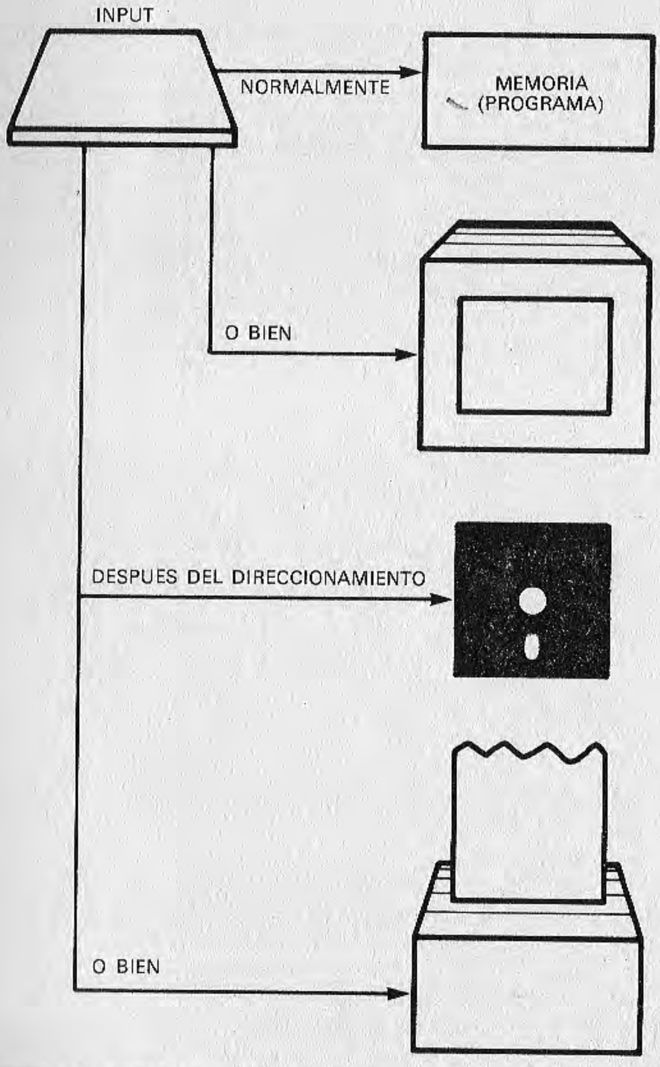


Figura 3.— Redireccionamiento de la entrada y salida de datos.

Por ejemplo, es posible pasar parámetros a un programa que se vaya a ejecutar sin emplear el teclado, haciéndole leer un fichero escrito por otro programa. O bien, utilizar el teclado, pero escribir directamente en la impresora en lugar de hacerlo en la pantalla. Para conseguirlo se han de utilizar los caracteres especiales "<" para la entrada y ">" para la salida.

• Transferencia de los datos. Permite utilizar los resultados de un programa, normalmente visualizados, como datos de entrada para otro programa que suele recibir los datos desde el teclado. Esta función de transferencia se denomina también "piping" porque el DOS no hace otra cosa sino "canalizar" los datos haciéndoles seguir un recorrido obligado (Fig. 4). Por ejemplo, si queremos que en la impresora aparezca la lista de todos los ficheros contenidos en un disco en un orden diferente del que nos proporciona el sistema operativo, podemos escribir:

```
DIR/SORT>PRN
```

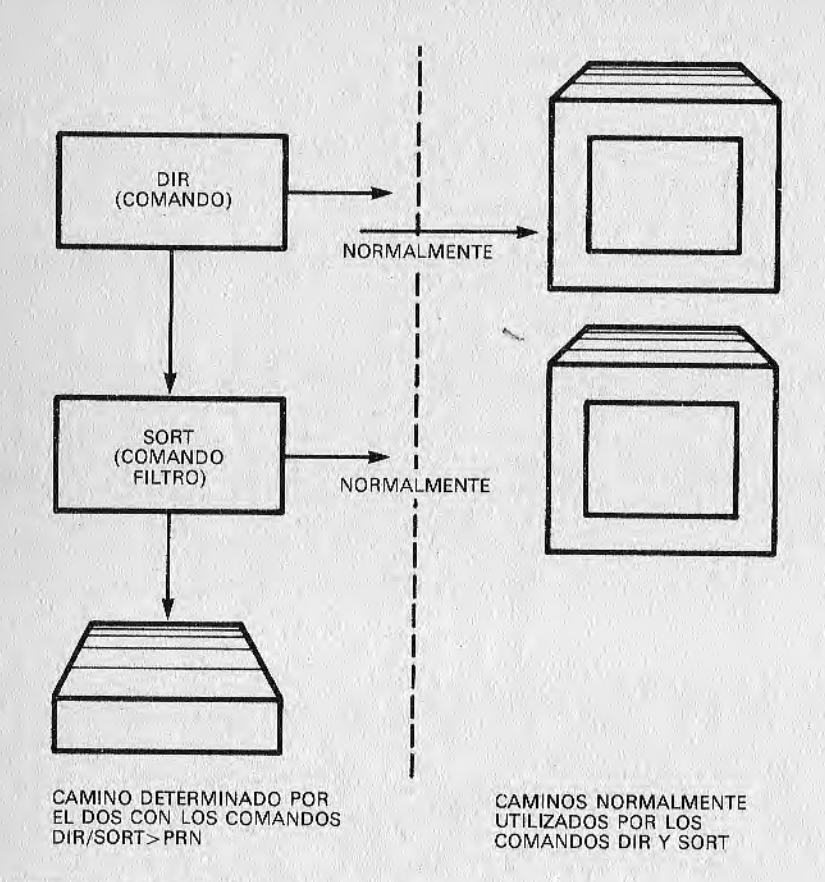
Con este comando, el DOS "canalizará" las informaciones de salida del comando DIR hacia el programa SORT que, en lugar de restituir los datos a la pantalla, los imprimirá.

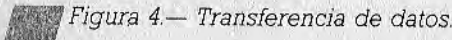
Todas estas nuevas posibilidades (sobre cuyos detalles se hablará más adelante) han permitido al MS-DOS, que se acogió con notable aceptación por los usuarios, llegar a ser un estándar para casi todos los ordenadores dotados de unidades CPU con los microprocesadores 8086 y 8088 de Intel. En los últimos años, como confirmación de dicha aseveración, muchos ordenadores vienen provistos del sistema operativo MS-DOS; constituyen la gama de los denominados "compatibles".

Cómo se gana el pan el MS-DOS

Tratemos de resumir las funciones del sistema operativo MS-DOS, que, son fundamentalmente:

- gestión, control y cuidado de la unidad central de proceso (CPU);
- intercambio de informaciones con las unidades periféricas (entrada/salida - E/S);
- gestión, control y empleo de la memoria central;
- gestión, control y utilización de las memorias de masa.




Figura 4.— Transferencia de datos.

La CPU ejecuta las instrucciones que se le suministran una tras otra y pueden realizar, por lo tanto, un solo trabajo cada vez, pero su velocidad de ejecución es tan alta que a menudo se encuentra inactiva porque quienes han de suministrar los datos a procesar no logran trabajar a la misma velocidad. ¡El componente más lento de todo el sistema es el hombre!

Para poner remedio a este inconveniente y eliminar gran parte de los tiempos muertos, se ha pensado en establecer los trabajos de modo que el procesador los pueda ejecutar uno tras otro sin interrupciones.

Este sistema de trabajo, con la introducción de los datos y comandos a ejecutar, se llama "batch processing" (proceso por "lotes") y permite optimizar el funcionamiento del ordenador.

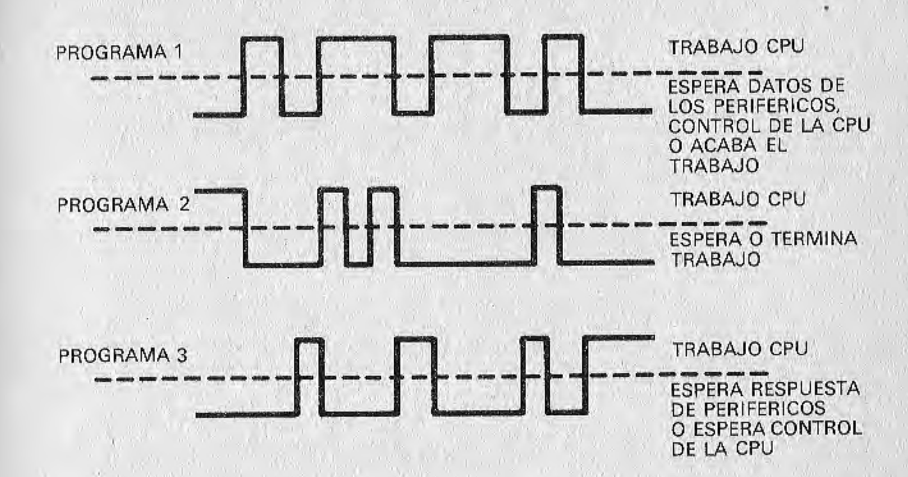


Figura 5.— Sistema Multitasking (multitarea). La CPU se ocupa simultáneamente en al menos dos programas con diferentes prioridades. Cada uno de los programas mantiene el control de la CPU hasta el momento en que tiene necesidad de datos procedentes de los periféricos. En la figura se representa una secuencia de trabajo. El número de programas que pueden trabajar simultáneamente no es indefinido, sino que depende, con frecuencia, de la cantidad de memoria disponible y del número de peticiones de Entrada/Salida de cada uno de los programas.

Otros DOS han adoptado soluciones, utilizadas con anterioridad en los grandes ordenadores. Una de ellas es la denominada "multitasking" o multitarea (por tarea se entiende aquí un trabajo). Con ella la CPU está ocupada simultáneamente en al menos dos trabajos. Los tiempos muertos de una tarea son aprovechados por un segundo programa, y los de este segundo programa, por un tercero o por el primero o, si no está todavía preparado, por el segundo, y así sucesivamente (Fig. 5). El sistema operativo, en este caso, está dedicado plenamente a la gestión y al control de todos los recursos a su disposición.

Por lo que respecta a la memoria central, se puede decir que el DOS está también aquí tanto más ocupado cuanto menos trabajo el programador quiera hacer. Es decir, si el programador quiere controlar directamente la organización de la memoria central, el DOS deja que lo haga, pero si, por el contrario, renuncia al control, entonces el DOS procede directamente a realizar todas las asignaciones necesarias sin olvidar ninguna.

CAPITULO II

PRIMEROS PASOS

Discos rígidos y flexibles

El MS-DOS está constituido, como se dijo anteriormente, por un conjunto de programas grabados en discos flexibles ("floppy disk" o disquetes). Por consiguiente, resulta imposible utilizarlo si no se posee, entre los diversos componentes del sistema, una unidad periférica que sea capaz de leer y grabar datos en estos soportes magnéticos. También es posible transferir el DOS desde el disco en el que se recibe, a otro soporte magnético formado por un material resistente, no flexible pero muy ligero, contenido dentro de una caja sellada. Este tipo de soporte, denominado disco rígido, puede montarse en el interior de la propia unidad central o bien exteriormente, pero no es posible extraerlo ni sustituirlo. En compensación, puede grabar muchos más datos que los disquetes y es más rápido y fiable. La unidad de disco flexible incorporada con el ordenador personal utiliza discos de 5 1/4 pulgadas (133 milímetros de diámetro). Este tipo de unidades permite la grabación de una sola o de ambas caras del disco.

El número de caracteres que es posible grabar en cada cara es 163.840 ó 184.320, según se utilice la versión 1.1 o 2.0 del MS-DOS. Teniendo, pues, a nuestra disposición unidades capaces de grabar ambas caras podemos almacenar, como máximo, en cada disco 327.680 ó 368.640 caracteres (320 ó 360 Kbytes, respectivamente).

Los discos, antes de utilizarse, deben ser "formateados", es decir, inicializados. Esta operación consiste en escribir informaciones en puntos determinados de las pistas que sirven para identificar el disco y como ayuda para el DOS.

En efecto, el comando correspondiente procede a:

- grabar el registro inicial ("boot") con el programa de partida;
- preestablecer el índice de los ficheros y la tabla de asignación de los mismos;
- identificar cada sector del disco.

Este comando de inicialización de los discos se denomina FORMAT. Con su ayuda se pueden copiar también en el disco el fichero que contiene el procesador de los comandos internos COMMAND.COM y los dos módulos BIOS y BDOS, que en la versión del IBM PC se denominan IBMBIO e IBMDOS, respectivamente.

Esta función permite, pues, crear un disco que contenga el MS-DOS y con espacio para los datos. Es preciso destacar aquí que el MS-DOS está constituido fundamentalmente por los tres ficheros mencionados, aunque dispone de muchos otros ficheros de comandos y programas de gran utilidad, pero no indispensables para el empleo del propio DOS.

Los "tres magníficos", que constituyen la base del sistema, ocupan unos 40.000 bytes en la versión 2.0 (13.000 en la antigua versión 1.1).

Se pueden utilizar discos formateados en la versión 1.1 incluso empleando la versión 2.0, pero no es posible lo contrario debido a la diferente disposición de los datos (la versión 2.0 tiene un sector más por pista, como veremos más adelante).

El formateado de un disco o, si lo prefiere así, su inicialización, no sirve solamente para grabar los tres ficheros base del DOS, sino también, y ante todo, para "dibujar" el disco, determinando la posición de todos las pistas y sectores que la cabeza de lectura/escritura puede explorar. El MS-DOS prepara los discos subdividiendo la superficie magnética en varios círculos concéntricos, denominados pistas, como si arrojara una piedra en un estanque (Fig. 1).

Todas las pistas están subdivididas en 8 (para la versión 1.0) o en 9 sectores (para la versión 2.0), cada uno de los cuales puede grabar 512 bytes (caracteres).

Organizadas de este modo, las informaciones se pueden extraer con gran facilidad por medio de tan solamente tres parámetros: cara, pista y sector. Este tipo de direccionamiento no es necesario si utilizamos las funciones que el DOS pone a nuestra disposición, a menos que, de forma excepcional, queramos controlar directamente las funciones de lectura y de escritura en lenguaje máquina, lo que es una tarea propia de verdaderos expertos.

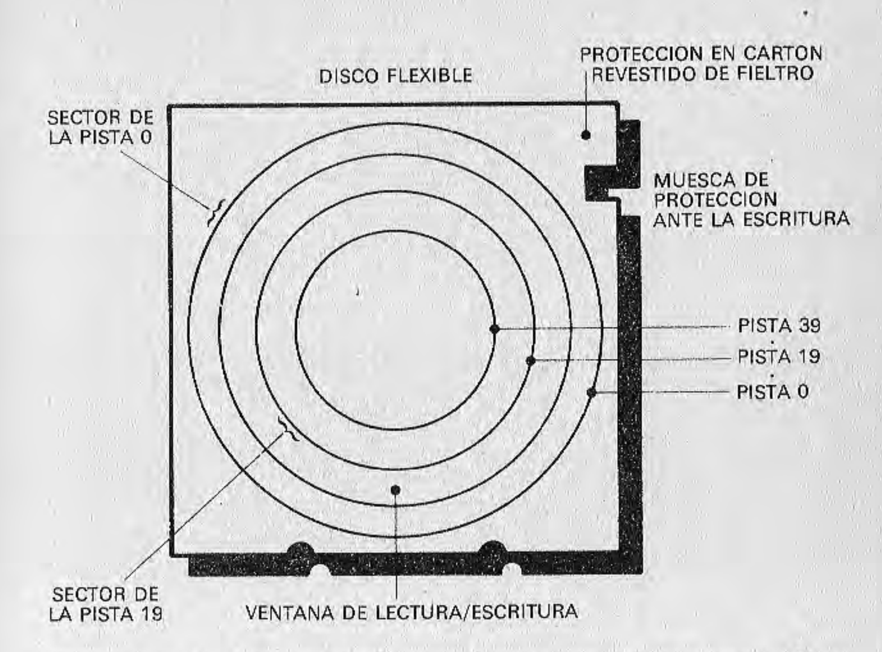


Figura 1.— Estructura del disco flexible una vez formateado. Si pudiéramos distinguir las pistas creadas por el formateo las veríamos semejantes a las ondas producidas por una piedra arrojada en un estanque. La pista con el número más alto es la más interna y la pista 0 es la más externa.

Cuidados y manejo de los discos flexibles

Los disquetes o discos flexibles son especialmente delicados, como deberíamos saber también por las instrucciones que suelen estar impresas en las fundas que los protegen del polvo.

Admitamos que les tratamos bien, pero ¿qué ocurrirá si, por error, escribimos sobre un disco que contiene ya datos y no hemos comprobado con anterioridad cuáles eran?

Hablando en términos generales, lo que sigue son las diversas posibilidades más o menos trágicas o afortunadas:

- los datos que hemos escrito (por ejemplo, el fichero A) no existían en el minidisco en cuestión, por lo que los datos nuevos se graban de forma normal (Fig. 2 - primer caso);
- en el disco no había espacio disponible suficiente para grabar otros datos y por ello no se grabaron; esta "desgracia" es indicada por el DOS (Fig. 2 - segundo caso);

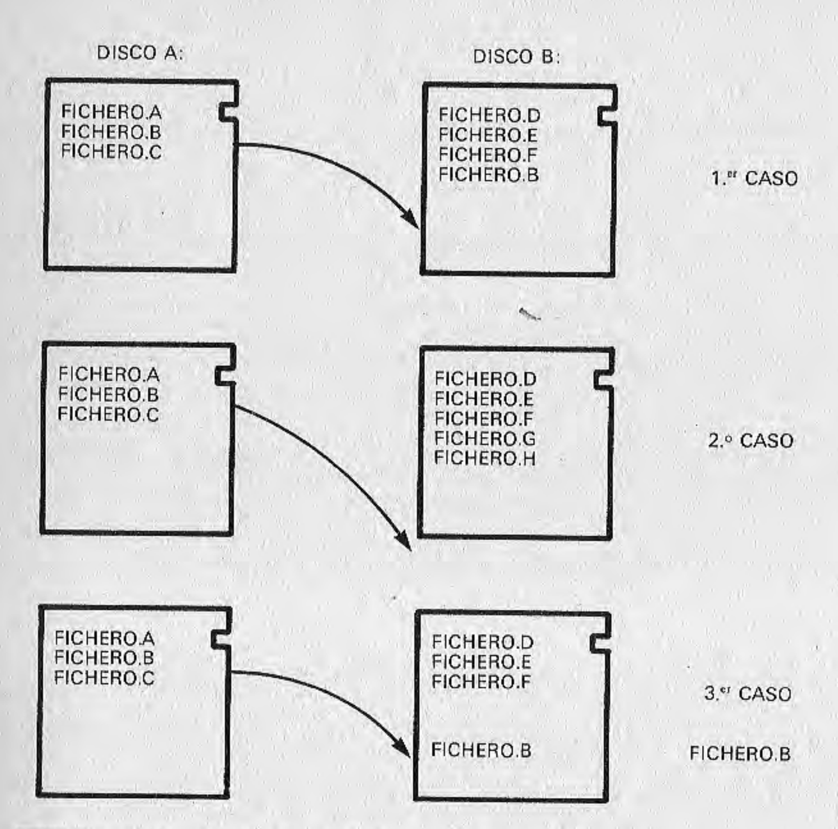


Figura 2.— Copia de un fichero sobre un disquete.

- en el disco existía un fichero con el mismo nombre que el utilizado para escribir los datos nuevos; en tal caso, podemos "decir adiós" a todo nuestro trabajo anterior porque el fichero nuevo ha sustituido al antiguo. Si hubiéramos hecho una copia de reserva del fichero podríamos recuperar todo lo que se borró. El DOS no advierte que está destruyendo el fichero ni tiene conocimiento de la existencia de un fichero anterior (Fig. 2 - tercer caso);
- en el caso de una inicialización con FORMAT o empleando el DISKCOPY, el contenido completo del disco estará irremediablemente borrado.

Para evitar que se pierdan tan fácilmente las preciosas informaciones contenidas en el disco, además de la técnica de la copia de reserva ("backup") se puede recubrir, con una etiqueta adecuada, la muesca lateral del disco flexible (que inhibe físicamente la escritura). Puede ser una medida oportuna para evitar algunos desastres.

Precisamente para evitar la pérdida accidental de los datos, algunos discos no poseen la muesca antes citada y, por ello, nunca se podrán grabar.

Los discos de una sola cara pueden ser leídos por una unidad de doble cara, pero no ocurre lo mismo a la inversa. Se pueden tener en la misma máquina dos unidades de discos diferentes: una que puede escribir y leer discos de doble cara, dotada de dos cabezas de grabación, y otra que puede grabar una cara al tener solamente una cabeza.

En un caso de esta naturaleza es necesario prestar mucha atención al manejo de los discos porque mientras que la unidad de doble cabeza puede leer y escribir también solamente una cara, no sucede lo contrario en la otra unidad.

Si hemos querido ahorrar (o nos ha parecido suficiente) la adquisición de una sola unidad, el DOS, quizás, porque considera injustificada dicha decisión, simula que se han instalado dos unidades y no una sola, con el resultado de hacernos perder un montón de tiempo para introducir y extraer los discos. Efectivamente, el DOS, cuando tiene a su disposición una única unidad de discos, simula la presencia de la segunda. Veamos un poco más detalladamente qué quiere decir esto.

Supongamos que tenemos dos unidades. Entonces, el sistema operativo de disco (DOS) las utilizará directamente dándoles las referencias de unidad A y unidad B; si damos la orden de copiar un fichero desde el disco existente en la unidad A al disco de la unidad B, el DOS ejecutará la orden sin chistar (Fig. 3).

Al tener a nuestra disposición una sola unidad ya no debemos pensar en términos de unidades, sino de discos. La orden se interpretará, pues, como:

Copia el fichero desde el disco asociado a la A al disco asociado a la B

Con este artificio lingüístico nos será sencillo seguir las instrucciones del DOS. Por ejemplo, probemos a escribir el comando:

A>COPY ARCHIVO.PAD B:

El sistema DOS lee el fichero ARCHIVO.PAD presente en el disco que está en ese momento introducido en la unidad (que, como se ve, es la A, pues aparece "A>"), lo guarda en memoria y escribe en la pantalla la solicitud:

Introducir disco en la unidad B: y pulsar cualquier tecla



CON DOS UNIDADES DE DISCO A SU DISPOSICION, EL DOS PUEDE UTILIZAR LA PRIMERA COMO UNIDAD DE DONDE LEE EL FICHERO Y LA SEGUNDA COMO AQUELLA QUE LO GRABA

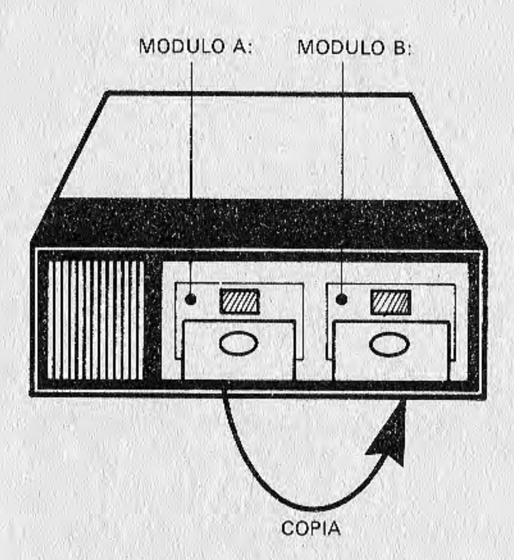


Figura 3.— Copia de un fichero desde un disco a otro cuando se dispone de dos unidades.

Es necesario obedecer; sacar el disco A e introducir el disco B (Fig. 4).

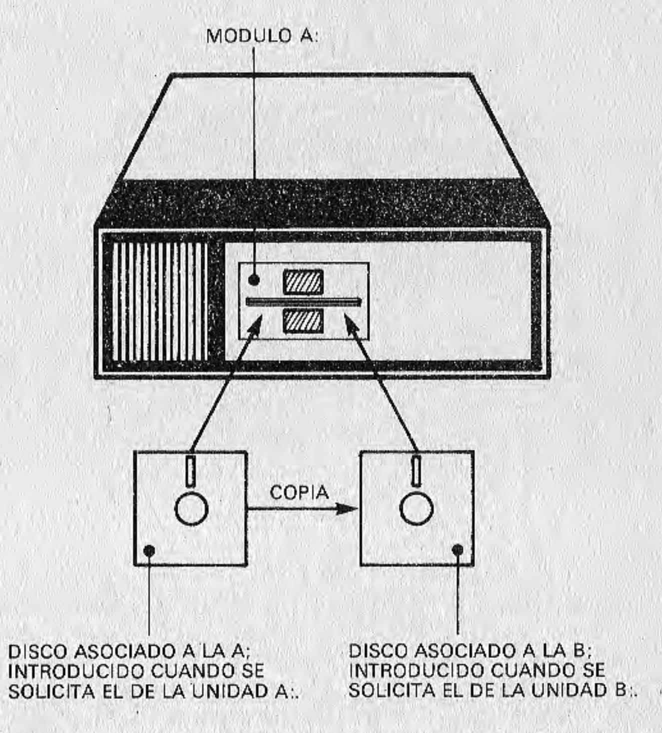


Figura 4.— Copia de un fichero desde un disco a otro disponiendo de una sola unidad. Cuando el DOS solicita el disco de la unidad B, es necesario extraer el que había (asociado a la A) e insertar otro, sobre el que grabaremos (asociado a la B).

Después de haber introducido este último y pulsar cualquier tecla, el DOS copiará el fichero sobre el disco B y aparecerá en la pantalla:

1 fichero copiado
A>

Acaba presentando una letra (de la unidad) y el signo de "mayor que" (>) que, como ya vimos en la monografía dedicada al CP/M, es el denominado "prompt", que indica el estado de espera de órdenes en que queda el DOS.

En este punto sabemos ya algunas nociones importantes:

- las unidades de disco para el sistema operativo DOS son siempre por lo menos dos; si existe una sola simulará la segunda;
- la letra que va seguida por el signo de petición de orden (en el ejemplo "A>") indica la unidad que supone por defecto;
- la letra de la unidad activa ("A>" o "B>") no es la de la última unidad utilizada; si observamos una vez más el comando que hemos escrito en el ejemplo veremos que después de su ejecución el DOS vuelve al disco que ha supuesto de partida. En efecto, después de haber copiado el fichero en el disco B, el DOS proporciona nuevamente el mensaje "A>", por cuanto la unidad A sigue siendo la activa, con independencia de que se haya ejecutado un trabajo en la unidad B. Este concepto es muy importante, pero volveremos a considerarlo más adelante.

En lo que respecta al disco rígido ("hard disk") puede utilizarse solamente con la versión 2.0 del MS-DOS. También este tipo de unidad de discos, capaz de contener ya no millares, sino millones de caracteres, debe formatearse para poder ser utilizado.

Hay que destacar el hecho de que la versión 2.0 del MS-DOS (PC-DOS) se ha traducido al español, por lo que, considerando que es bastante más cómodo que traducir del inglés, tomaremos en consideración solamente esta versión.

En el caso de que tuviéramos un disco duro en nuestro ordenador (los suministrados por el fabricante suelen contener 10 millones de caracteres), tendremos la posibilidad de utilizarlo de muy diversas maneras.

Una de las características más interesantes de este tipo de unidades de disco (rígido) es la de poder subdividirse en diversas secciones, denominadas "particiones"). Estas particiones, hasta un máximo de cuatro, pueden ser de diferente amplitud y cada una de ellas puede ser activada por un sistema operativo distinto.

En la práctica, si se quieren utilizar sistemas operativos diferentes con las unidades de disco flexible basta introducir el disco que contiene el sistema operativo que nos interesa en ese momento; no se puede hacer lo mismo con el disco duro, pues es muy difícil e incómodo desmontarlo para volver a montar otro.

En caso de que se conozcan previamente cuántos y cuáles sistemas operativos se utilizarán, será posible subdividir el disco rígido basándose en dichas exigencias.

También el disco duro se identifica por el procesador mediante letras; cuando el DOS se pone en marcha identifica las unidades de disco flexible con A y B (hemos visto que utiliza la B aunque tengamos una sola unidad física de disco) y con C a la unidad de disco duro (si somos afortunados y tenemos una segunda, le correspondería la D).

Llegó el momento de utilizar el DOS

Armémonos de valor y "entremos" por fin en el DOS. Ante todo aprendamos cómo se efectúa la carga, por primera vez, del propio DOS en el ordenador, de modo que pueda servirnos con su habitual fidelidad.

Abramos el paquete que hemos llevado a casa, después de haber instalado nuestro ordenador, y procedamos a extraer el DOS fuera de su atractiva envoltura, constituida por una caja de cartón rígido muy resistente.

Retiremos con cuidado la tapa: encontraremos, perfectamente embalada, una carpeta de tres anillas que, además de las instrucciones, contiene, en unas bolsas adecuadas, dos discos. El método de presentación con hojas sueltas es muy cómodo, tanto para la actualización como para la consulta. Saquemos de su funda de plástico el disco del MS-DOS.

Este disco no es directamente utilizable como sistema operativo: todas las veces que lo insertemos nos solicitará que hagamos una copia de él y que la "personalicemos"; en efecto, lo mejor que podemos hacer es precisamente realizar una copia del disco original y luego guardarlo para poder disponer del mismo con posterioridad.

Antes de conectar el ordenador, encendamos todos los periféricos; luego busquemos a la derecha de la unidad central el interruptor de alimentación y pongámoslo en su posición de encendido. Después de haberlo comprobado por sí mismo, el procesador tratará de cargar el sistema operativo a partir de la unidad A; si el disco del sistema operativo está introducido en la unidad A y la puerta de protección está cerrada, el disco comenzará a girar, se encenderá el piloto rojo de la unidad y el procesador leerá, llevándolos a la memoria, los tres programas base del DOS. Si, por el contrario, el disco no está introducido en la unidad o la puerta no está cerrada, el procesador cargará la versión del BASIC para casete contenida en su memoria ROM; aparecerá en la pantalla el mensaje siguiente:

```
The IBM Personal Computer Basic.
Versión C1.10 Copyright IBM Corp 1981
62940 Bytes free

OK
—
```

en este caso, "OK" es el indicador de presencia ("prompt") del BASIC y nos confirma que estamos en el ámbito adecuado.

Deberíamos haber introducido el disco antes de comenzar, pero no nos preocupemos, porque no es preciso cortar la alimentación. Hay otro sistema para conseguir la inicialización del sistema sin recurrir al interruptor y sin tener que rehacer todos los controles iniciales (la espera es directamente proporcional a la cantidad de memoria instalada; cuanto más memoria haya tanto más largos serán los controles). Introduzcamos en la unidad A el disco original del MS-DOS, cerremos la puerta y pulsemos simultáneamente las teclas CTRL, ALT (a la izquierda del teclado) y DEL (en la parte inferior derecha). Soltamos ahora las tres teclas, y veamos lo que sucede.

El piloto de la unidad A se iluminará y se oirá el desplazamiento de la cabeza de lectura. En este preciso momento, el procesador comienza a cargar el DOS.

Una vez terminada la carga, entra inmediatamente en actividad el procesador de las órdenes o comandos, que hemos considerado como tercer componente del DOS. Ante todo, procederá a buscar en el disco a partir del cual se ha cargado el sistema DOS el fichero AUTOEXEC.BAT. Si existe este fichero, el procesador lo leerá y ejecutará todas las órdenes que estén grabadas en el mismo.

El sufijo (extensión) BAT es la abreviatura de la palabra inglesa BATCH (Lotes) que vimos al hablar de la posibilidad de realizar varias tareas a la vez en el ordenador con el fin de eliminar los tiempos muertos.

Este fichero es muy importante porque es inmediatamente ejecutado por el procesador cada vez que se inicializa el sistema DOS. En consecuencia, se leerá el fichero AUTOEXEC.BAT contenido en el disco original del DOS y, una vez realizada dicha lectura, comenzará la ejecución de los comandos que contiene. Esta serie de comandos está, en este czaso, preestablecida, pero veremos a continuación cómo podemos introducir por nuestra cuenta en el fichero AUTOEXEC.BAT la serie de comandos que nos interesa particularmente.

Para simplificar las cosas, de ahora en adelante supondremos que disponemos de dos unidades de discos y así trataremos de crear una copia "personalizada" del DOS. El término "personalizada" quiere decir capaz de aceptar los caracteres enviados desde el teclado, nacional o extranjero, que hemos elegido al hacer el pedido del ordenador, que tendrá las teclas y los caracteres propios del idioma que hemos elegido utilizar.

El MS-DOS, después de haber leído los comandos contenidos en el fichero AUTOEXEC.BAT, comenzará a ejecutarlos y aparecerá en la pantalla el primer mensaje:


```
Cargada parte residente comando MODE
A>REM PARA OBTENER COPIA EN ESPAÑOL DEL
A>REM DOS, EJECUTAR LOS PASOS SIGUIENTES
.............................................
A>REM PASO 1
A>REM Mantener el disco DOS original
A>REM en la unidad A y luego, procurarse
A>REM un segundo disco nuevo
A>DISKCOPY A: B:
Introducir disco original en la unidad A:
Introducir disco destino en la unidad B:
Pulsar una tecla cualquiera
```

La palabra REM (idéntica a la del BASIC, pero no hemos de olvidar que estamos en el DOS) es la abreviatura de la palabra inglesa REMark y se utiliza para indicar las líneas de comentario o de advertencias.

El MS-DOS nos pide ahora que introduzcamos el disco original en la unidad A y el disco virgen, que está destinado a recibir la copia, en la unidad B. Al pulsar una tecla cualquiera se iniciará la ejecución del comando DISKCOPY que crea una copia del disco original; si el disco que ha de contener la copia no está todavía preparado para la grabación, procederá a efectuar el formateado.

Una vez terminada la copia del disco, aparecerá el mensaje:

```
Copia terminada
Otra copia (S/N)?
```

De este modo, el sistema operativo trata de saber si queremos efectuar otra operación de copia del disco.

Si quisiéramos efectuar otra copia bastaría responder con la letra S para repetir la operación que acabamos de realizar y, en caso contrario, responderíamos con N. El sistema operativo nos advierte que hemos terminado el primer paso:

```
A>REM PASO 2
A>REM El disco copia se compara ahora
A>REM con el disco DOS original
A>DISKCOMP A: B:
Introducir el primer disco en la unidad A:
Introducir el segundo disco en la unidad B:
Pulsar una tecla cualquiera
```

El DOS, después de haber efectuado la copia del disco, compara el original y la copia para ver si todo se hizo de forma correcta y si todos los caracteres se han transferido al lugar adecuado.

Si la comparación fue satisfactoria se podrá proseguir con el paso sucesivo, y, de no ser así, nos veremos obligados a repetir las operaciones anteriores. Podemos volver a comenzar desde el principio en una de las dos formas que hemos aprendido:

- poner el interruptor en la posición de apagado, esperar unos segundos y volver a encender;
- pulsar las tres teclas ALT, CTRL y DEL de forma simultánea.

Pero ahora existe un tercer modo de reiniciar la operación, puesto que el sistema operativo se ha cargado ya y, por consiguiente, podemos dar la primera orden propiamente dicha.

Si introducimos la palabra AUTOEXEC veremos que se produce una inicialización de todo el conjunto sin intervención alguna por nuestra parte. En este caso, somos nosotros los que damos la orden de comienzo de la ejecución de los comandos contenidos en el fichero AUTOEXEC y no el DOS. Si también en esta segunda ocasión nos encontramos con que el proceso no es correcto, y estamos seguros de haber ejecutado al pie de la letra cuanto se nos ha pedido, probemos a sustituir el disco que ha de recibir la copia y procedamos a una nueva inicialización; en efecto, es posible que estemos utilizando un disco defectuoso.

Si también después de la sustitución no obtenemos resultados satisfactorios es muy probable que haya algo que no funcione adecuadamente en el sistema y será conveniente consultar los "procedimientos de diagnóstico" contenidos en el manual "Guía del Usuario del Ordenador Personal IBM". Afortunadamente, es muy poco frecuente que encontremos estas dificultades, pero siempre es mejor saber lo que nos puede suceder en algunos casos extremos.

Para seguir adelante como dice el mensaje es suficiente pulsar una tecla cualquiera. Hagámoslo y veremos aparecer en la pantalla un nuevo mensaje:

```
A>REM PASO 4
A>REM Guardar el disco con el DOS
A>REM original en su funda
A>REM de protección. Introducir el disco con el
A>REM DOS en español en la unidad A
A>PAUSE
Pulsar una tecla cualquiera
```

Si todavía no lo hemos hecho, es la ocasión propicia para pegar una etiqueta autoadhesiva en el disco y escribir en ella: "DOS español". Tratemos de conservar con cuidado el disco original

porque, si perdiésemos o tuviéramos que modificar la copia traducida, siempre podríamos volver a obtener otra copia.

Supongamos que hemos introducido el DOS traducido en la unidad A. Pulsemos una tecla cualquiera y aparecerá en la pantalla un nuevo mensaje:

```
1=Americano(USA)  2=Francés  3=Alemán  4=Italiano
5=Español  6=Inglés  0=Salida
```

Podemos elegir entre estas posibilidades el idioma que preferimos utilizar. Es necesario haber elegido antes el tipo de teclado correspondiente, es ilógico, si no se trata de un caso excepcional, haber solicitado un teclado español y utilizar la "personalización" de dicho teclado con los caracteres americanos o franceses.

En este punto, el sistema DOS le deja con las operaciones extraordinarias y prosigue con las que suelen realizar cuando se le inicializa. Los mensajes

```
A>KEYBSP
A>WTDATIM
Introducir la fecha (DD-MM-AA):01-01-1986
Modificación de la fecha
```

y todos los que sigan los veremos aparecer todas las veces que activemos el sistema DOS. Ahora, la copia traducida está completa y nos convendrá protegerla tapando la muesca del disco.

El DOS nos está pidiendo ahora, después de indicarnos que ha adoptado el teclado español, que introduzcamos la fecha o que confirmemos la que nos ha indicado en el mensaje.

Volvemos a introducir el disco con la copia traducida del DOS en la unidad A, cerramos la puerta de protección y escribimos la fecha exactamente en el modo que se nos ha pedido o bien damos un ENTER (pulsando la tecla) para confirmar que la fecha indicada es la correcta.

Hasta este momento no hemos podido escribir nada "autónomamente" en la pantalla, no hemos hecho otra cosa que pulsar teclas o responder de forma afirmativa o negativa a las preguntas, pero a partir de ahora podremos ver lo que escribimos visualizado en la pantalla y, además, observaremos también la presencia del cursor. Su continuo parpadear parece solicitarnos datos de continuo.

La fecha es importante en el MS-DOS porque se utiliza para grabarla en el índice de los ficheros junto a cada nombre, tanto en el momento de la creación de dicho fichero como al efectuar una modificación ocasional.

Podemos introducir la fecha utilizando las teclas numéricas de la fila superior del teclado; la podemos escribir de muchos modos (8-9-86, 8/9/86, 8/09/86, 08/9/1986, 08-09-86).

El MS-DOS realiza las comprobaciones habituales de mes, día y año y si los datos introducidos no son válidos lo indicará volviendo a solicitar la escritura de la fecha.

En el supuesto de que hayamos superado este escollo nos pedirá inmediatamente que introduzcamos la hora, también grabada en el índice de los ficheros (en ciertos casos en los que se dispone de tarjetas de expansión con un reloj y batería incluidos basta pulsar ENTER para que asuma la hora y fecha de la tarjeta).

Algunas aplicaciones utilizan la fecha y la hora grabadas y, por consiguiente, es mejor acostumbrarse a introducir de forma correcta estos datos.

La hora debe ser suministrada en el formato.

HORA:MINUTOS:SEGUNDOS

lo cual quiere decir que la separación entre las cifras debe estar ocupada por dos puntos (:). La única variante posible es la no introducción de los segundos, en cuyo caso el DOS, por defecto de información, supondrá el valor cero. También la hora será objeto de comprobación, si "supera el examen" se verán en la pantalla los mensajes siguientes:

```
A>REM The IBM Personal Computer DOS
A>REM Version 20 (C) Copyright IBM Corp 1981, 1982, 1983
A>
A>-
```

El último mensaje es, como se dijo anteriormente, el mensaje de petición ("prompt") del sistema DOS, que invita al usuario a introducir comandos u órdenes.

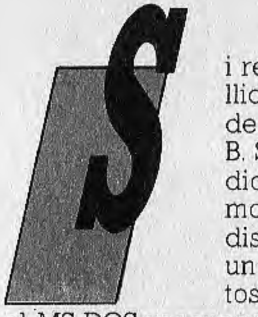
Además, nos comunica, con la letra que precede al carácter ">", con qué unidad de disco trabajamos en ese momento. Por ejemplo, si el mensaje es "A>" querrá decir que la unidad actualmente supuesta es la unidad A, y si el mensaje es "B>", que la unidad supuesta es B.

Con lo anterior hemos concluido la carga del sistema operativo y estamos en condiciones de proceder a introducirnos en los primeros comandos.

CAPITULO III

NOMBRES DE LOS FICHEROS Y COMANDOS MÁS UTILIZADOS

Nombre, apellidos y dirección de un fichero



i recuerda, hemos escrito ya el "nombre y apellidos" de un fichero cuando abordamos el tema de copiar un fichero desde el disco A al disco B. Servía también, de manera implícita, para indicar en dónde residen los ficheros: en las memorias de masa y, en ese caso concreto, en un disco. También debe haber quedado claro que un fichero es un programa o un archivo de datos y que tiene un nombre que lo identifica; así, el MS-DOS reconoce el fichero propio por medio de su nombre. Deberemos, pues, llamar a nuestros ficheros con un nombre; pero frecuentemente el nombre sólo no basta para identificar de forma unívoca a un fichero.

Efectivamente, imaginemos que hemos grabado en un disco una serie de ficheros, cada uno de ellos conteniendo las informaciones relativas a todos nuestros amigos. Tenemos así una serie de ficheros tales como:

BERNARDO
LUIS
NICOLAS
ROMULO

Si un cierto día se añade a nuestra agenda el nombre de LUCIANO como otro nuevo amigo no habrá ningún problema porque no existe dicho nombre en la lista y por ello es suficiente agregar el nombre del recién llegado, pero ¿qué sucede si nos hacemos amigos de otra persona también llamada LUIS?

Una solución podría consistir en introducirlo en otro disco, pero sería un poco incómodo porque no hay límites y el círculo de nuestros amigos podría enriquecerse con otros que tuvieran también el nombre de LUIS y cada nuevo amigo precisaría otro disco. Para evitar este inconveniente se ha inventado el "apellido" que para el MS-DOS equivale a lo que se denomina "extensión": forma parte integrante de la especificación de un fichero. La extensión está constituida por un máximo de tres caracteres que, precedidos por un punto, siguen al nombre de los ficheros, tal como se indica en los ejemplos siguientes:

BERNARDO.AAA
LUIS.AAA
LUIS.BBB
NICOLAS.AAA
ROMULO.AAA

Tanto para nosotros como para el sistema operativo es de gran utilidad poder identificar a los ficheros también en base a la extensión; y no solamente cuando el nombre es idéntico, sino también cuando, al tener ficheros con nombres muy diferentes entre sí, queremos identificarlos para reagruparlos o reconocerlos a primera vista (por ejemplo, dar la extensión "BAS" a todos los programas escritos en BASIC...).

A continuación damos ejemplos de utilización de las extensiones. Son extensiones estándar a las que el MS-DOS u otros programas de aplicación, asignan un significado bien concreto:

Extensión	Tipo	Ejemplo
COM	Utilizado para ficheros que contienen comando o programa.	COPY.COM DISKCOPY.COM
EXE	Se emplea para identificar ficheros que puedan ser ejecutados.	EDIT.EXE
OBJ	Identifica ficheros que contienen el resultado de compilaciones.	MOD1.OBJ MOD5.OBJ
MAP	Suele contener las listas de los índices de los discos y las tablas obtenidas por el programa LINK.	DSK1.MAP PROG.MAP
LIB	Utilizado por los compiladores para identificar las diversas bibliotecas.	USERLIB.LIB BASRUN.LIB.
TMP	Empleado para ficheros creados y anulados por otros programas.	EDIT.TMP VM.TMP

Extensión	Tipo	Ejemplo
BAT	Sirve para ficheros que contengan una serie de comandos del DOS (Ficheros por lotes)	AUTOEXEC.BAT SALVA.BAT
ASM	Fichero fuente en lenguaje ensamblador.	PROG1.ASM PROG2.ASM
BAS	Fichero fuente en lenguaje BASIC.	PROG3.BAS PROG4.BAS
HEX	Individualiza programas en lenguaje máquina (hexadecimal).	PROG5.HEX PROG6.HEX
BAK	Extensión de ficheros creados por programas de edición: se trata de ficheros de reserva o "backup".	ALFA.BAK
LST	Ficheros preparados para imprimirse con el comando PRINT.	PROSP1.LST PROSP2.LST
TXT	Ficheros que contienen los textos editados antes de darlos como elemento de trabajo a diversos programas.	CARTA.TXT TEXTO.TXT
HLP	Suelen contener los textos de las pantallas de ayuda en las aplicaciones.	PASO1.HLP PASO2.HLP
PRF	Contienen características que "personalizan" algunos dispositivos o programas.	EPSON.PRF GRAF.PRF

Estas son solamente algunas de las posibles abreviaturas útiles para identificar inmediatamente el tipo del fichero.

Hemos visto así el "nombre y apellidos" de un fichero, pero, como las personas, también los ficheros tienen una "casa" en donde residen, tal como el caso de los discos. Estos últimos toman su nombre de la unidad en la que están incluidos. Recordemos también que hay, como mínimo, dos unidades de discos (la A y la B) y, por consiguiente, si queremos llamar a un fichero con todos los elementos que lo identifican tendremos que escribir:

A:LUCIANO.AAA o bien B:LUCIANO.AAA

La letra que precede al nombre del fichero, seguida siempre por dos puntos, sin ningún espacio intermedio, individualiza pre-

cisamente la unidad en la cual está introducido el disco con el fichero en cuestión.

Esta especificación debe escribirse siempre a no ser que el fichero resida en un disco introducido en la unidad supuesta por el MS-DOS en ese momento.

Supongamos que el DOS esté a la espera de órdenes o comandos y que visualice en la pantalla:

B>

en este caso, si el fichero que buscamos está en el disco insertado en la unidad B no hay necesidad de especificar la unidad. Bastaría:

LUCIANO.AAA

Si, por el contrario, el fichero está en la unidad A nos veremos obligados a especificar:

A:LUCIANO.AAA

Hemos de recordar que siempre es posible cambiar la unidad supuesta (o de defecto). Basta introducir la letra identificadora de la nueva unidad seguida por dos puntos:

B>A: pulsar la tecla de introducción;
A> resultado

A continuación se dan una serie de normas que deberán observarse siempre:

- el nombre de un fichero no puede tener una longitud superior a ocho caracteres;
- pueden utilizarse los números, las letras del alfabeto y los caracteres especiales siguientes:

$#&@!%()—{}'—'

En las versiones anteriores a la versión 2.0 era posible utilizar también los caracteres

|<>\

- la extensión, precedida por un punto, no puede tener más de tres caracteres y sigue las reglas del nombre para los caracteres especiales;

- no se admiten espacios entre los caracteres;
- no se admiten las comas;
- no puede faltar el nombre del fichero o, lo que es lo mismo, es imposible encontrar un fichero.BAS, pero puede existir solamente el nombre sin la extensión.

Si el nombre y la extensión de un fichero respetan todas estas reglas, el DOS acepta el conjunto de caracteres y asocia la información del fichero solamente con ese conjunto de caracteres. Si queremos llamarlo debemos utilizar los mismos caracteres que proporcionamos la primera vez.

Las tres partes que individualizan a un fichero (indicación de la unidad, nombre del fichero y extensión), se denominan, en conjunto, la "especificación del fichero".

Finalmente, para seguir avanzando, vamos a citar los comandos más frecuentemente utilizados.

DIR, el índice o directorio

Si hemos seguido las indicaciones dadas hasta este punto deberemos tener en la unidad A el disco del DOS. Veamos cómo se llaman los ficheros que están contenidos en este disco. Para obtener todas las indicaciones de los ficheros escribimos el comando DIR, que produce en la pantalla una serie de informaciones que se desplazan rápidamente hacia arriba. Cuando la pantalla está llena, las líneas que siguen apareciendo por la parte inferior de la pantalla "empujan" fuera del campo visual a las que se encuentran en la parte superior. Este comando es la abreviatura de la palabra inglesa "Directory", que literalmente quiere decir lista de nombres o directorio. También se denomina índice de los ficheros o, simplemente, con la traducción literal "directorio". Permite comprobar cuántos y cuáles son los ficheros contenidos en un disco. Basta escribir DIR y dar al ENTER para obtener esta información.

Podemos obtener también el índice de los ficheros del disco en la unidad B con la simple introducción de:

A>DIR B:

El desplazamiento de las líneas hacia arriba ("scroll") ha concluido y tendremos ante nosotros una presentación visual en la pantalla del tipo siguiente:

MODE	COM	3199	31-10-85	12:00p
FDISK	COM	6321	31-10-85	12:00p
BACKUP	COM	3740	31-10-85	12:00p

RESTORE	COM	4155	31-10-85	12:00p
PRINT	COM	4627	31-10-85	12:00p
RECOVER	COM	2412	31-10-85	12:00p
ASSIGN	COM	826	31-10-85	12:00p
TREE	COM	1572	31-10-85	12:00p
GRAPHICS	COM	789	31-10-85	12:00p
SORT	EXE	1376	31-10-85	12:00p
FIND	EXE	5814	31-10-85	12:00p
MORE	COM	4386	31-10-85	12:00p
BASIC	COM	16256	31-10-85	12:00p
BASICA	COM	25984	31-10-85	12:00p
KEYBUK	COM	1221	31-10-85	12:00p
KEYBFR	COM	1669	31-10-85	12:00p
KEYBSP	COM	1541	31-10-85	12:00p
KEYBIT	COM	1285	31-10-85	12:00p
KEYBGR	COM	1573	31-10-85	12:00p
KBDOS	BAS	2027	31-10-85	12:00p
KBPGM	BAS	4900	31-10-85	12:00p
WTDATIM	COM	1540	31-10-85	12:00p
GRAFTABL	COM	1092	31-10-85	12:00p

33 Archivo(s) 1536 bytes libres

Antes de indicar cómo es posible ver otras informaciones relativas a los ficheros que hemos entrevisto fugazmente, dediquemos unos instantes a estudiar la presentación visual que tenemos delante en este momento. Observamos, a la izquierda, las especificaciones de los ficheros en las dos primeras columnas, con el nombre y la extensión (si existe) y luego, más a la derecha, el número de caracteres que cada fichero ocupa en el disco, seguido por la fecha y la hora en la que se grabó el fichero.

La última línea indica el número de ficheros grabados y el número de bytes libres a nuestra disposición en el disco.

Tratemos ahora de ver el comienzo de este directorio de 33 ficheros o archivos, de los cuales sólo vimos los 23 últimos. Para hacerlo, introducimos el comando:

DIR A:/P

El parámetro /P produce una interrupción del "scroll" (desplazamiento hacia arriba de las líneas) cuando la pantalla está llena y, en consecuencia, lo que se nos presentará ahora será lo siguiente:

El volumen en la unidad A no tiene etiqueta
Directorio de A:

COMMAND	COM	18272	31-10-85	12:00p
AUTOEXEC	BAT	1262	31-10-85	12:00p
ANSI	SYS	1664	31-10-85	12:00p
FORMAT	COM	7067	31-10-85	12:00p
CHKDSK	COM	6618	31-10-85	12:00p
SYS	COM	1416	31-10-85	12:00p
DISKCOPY	COM	2561	31-10-85	12:00p
DISKCOMP	COM	2207	31-10-85	12:00p
COMP	COM	2653	31-10-85	12:00p
EDLIN	COM	4509	31-10-85	12:00p
MODE	COM	3199	31-10-85	12:00p
FDISK	COM	6321	31-10-85	12:00p
BACKUP	COM	3740	31-10-85	12:00p
RESTORE	COM	4155	31-10-85	12:00p
PRINT	COM	4627	31-10-85	12:00p
RECOVER	COM	2412	31-10-85	12:00p
ASSIGN	COM	826	31-10-85	12:00p
TREE	COM	1572	31-10-85	12:00p
GRAPHICS	COM	789	31-10-85	12:00p
SORT	EXE	1376	31-10-85	12:00p
FIND	EXE	5814	31-10-85	12:00p
MORE	COM	4386	31-10-85	12:00p
BASIC	COM	16256	31-10-85	12:00p

Pulsar cualquier tecla para continuar.

y si pulsamos una tecla cualquiera, el "scroll" continuará con:

BASICA	COM	25984	31-10-85	12:00p
KEYBUK	COM	1221	31-10-85	12:00p
KEYBFR	COM	1669	31-10-85	12:00p
KEYBSP	COM	1541	31-10-85	12:00p
KEYBIT	COM	1285	31-10-85	12:00p
KEYBGR	COM	1573	31-10-85	12:00p
KBDOS	BAS	2027	31-10-85	12:00p
KBPGM	BAS	4900	31-10-85	12:00p
WTDATIM	COM	1540	31-10-85	12:00p
GRAFTABL	COM	1092	31-10-85	12:00p

33 Archivo(s) 1536 bytes libres

Como puede observar, ahora tenemos la posibilidad de comprobar todo el índice y de detenernos en cualquiera de sus partes hasta que no decidamos seguir adelante.

Observamos también que el comando DIR proporciona otras informaciones al comienzo del índice o directorio. Una es la relativa a la unidad en la que está introducido el disco en cuestión y la otra es el nombre del disco utilizado, siendo posible, en el momento de la inicialización, dar un nombre al disco.

La barra inclinada invertida (\) que aparece detrás de "A:" nos

indica que el directorio es raíz y no un subdirectorio, como luego veremos.

Continuamos tratando algunas características de los nombres de los ficheros y del comando DIR.

Existen dos caracteres, el signo de interrogación y el asterisco, que, en el ámbito de la especificación de los ficheros, tienen significados especiales; se denominan caracteres globales o de enmascaramiento. Para aprender a utilizarlos es conveniente dar un par de ejemplos prácticos. Probemos a introducir el comando

A>DIR*.BAS.

Veremos aparecer en la pantalla la lista de todos los ficheros que tienen en la especificación del fichero la extensión BAS:

El volumen en la unidad A no tiene etiqueta.
Directorio de A:\

KBDOS	BAS	2027	31-10-85	12:00p
KBPGM	BAS	4900	31-10-85	12:00p

2 Archivo(s) 1536 bytes libres

Probemos ahora con

A>DIR KEY*.*

Esta vez aparecerá la lista de todos los ficheros que tienen su nombre comenzando con las letras KEY y luego prosiguen con algún otro carácter y extensión.

El volumen en la unidad A no tiene etiqueta.
Directorio de A:\

KEYBUK	COM	1221	31-10-85	12:00p
KEYBFR	COM	1669	31-10-85	12:00p
KEYBSP	COM	1541	31-10-85	12:00p
KEYBIT	COM	1285	31-10-85	12:00p
KEYBGR	COM	1573	31-10-85	12:00p

5 Archivo(s) 1536 bytes libres

Esencialmente, el carácter del asterisco proporciona al DOS la información de "cualquier número y tipo de caracteres en mi lugar". En el último ejemplo, esta información se da tanto en el nombre como en la extensión.

Probemos otros ejemplos:

DIR A:*.E*
DIR A:AUTO*.*

El otro carácter global, el signo de interrogación, proporciona al DOS la información de "cualquier único carácter en mi lugar".

Veamos inmediatamente un ejemplo ilustrativo:

A>DIR KEY?I?.*

La lista que tenemos ahora en la pantalla nos indica que tenemos un solo fichero que responda a nuestra solicitud.

El volumen en la unidad A no tiene etiqueta.
Directorio de A:\

KEYBIT	COM	1285	31-10-85	12:00p

1 Archivo(s) 1536 bytes libres

Podemos también combinar los dos caracteres en el mismo comando:

A>DIR ??S*.C*

El volumen en la unidad A no tiene etiqueta.
Directorio de A:\

DISKCOPY	COM	2561	31-10-85	12:00p
DISKCOMP	COM	2207	31-10-85	12:00p
RESTORE	COM	4155	31-10-85	12:00p
ASSIGN	COM	826	31-10-85	12:00p
BASIC	COM	16256	31-10-85	12:00p
BASICA	COM	25984	31-10-85	12:00p

6 Archivo(s) 1536 bytes libres.

Como puede observar no se ha listado el fichero ANSI.SYS que tiene la S en la tercera posición del nombre, pero que no responde a la solicitud de tener la letra C en la primera posición de la extensión.

Ahora volvamos a tratar un comando al que se hizo alusión con anterioridad.

FORMAT, el arado que hace el surco

Ante todo, comencemos por preparar un disco para poder hacer nuestras pruebas. Hemos visto que el comando que inicializa el disco, es decir, aquel que le prepara para recibir los datos y que establece el índice de los ficheros, es el comando FORMAT. Comprobemos que en nuestra pantalla aparece ahora el indicador de presencia y de petición de datos del DOS:

A>

Si no apareciera, le recordamos los dos procedimientos de partida: pulsar el interruptor de red o bien CTRL y la pulsación simultánea de las teclas ALT y DEL.

Una vez obtenido el mensaje A> podemos introducir el comando FORMAT o, en letras minúsculas, format o también ForMaT. Las letras mayúsculas y minúsculas no establecen diferencia alguna porque el MS-DOS las transforma todas ellas a mayúsculas. Una vez escrito el comando tendremos que pulsar también la tecla de introducción (Enter o Return).

La tecla de introducción da la "señal de salida" al procesador de los comandos, que comenzará a interpretar el comando para llevar a cabo su ejecución inmediata. Si observamos la luz roja de la unidad de disco A: observaremos que al pulsar la tecla de introducción se enciende por un instante. Ello significa que el DOS está leyendo un comando que reside en el disco. El comando está contenido en un fichero que se denomina FORMAT.COM y que se carga en memoria, cediendo el DOS el control al FORMAT que, para comenzar, emitirá un mensaje:

Introducir disco nuevo en unidad A:
y pulsar cualquier tecla.

Ejecutemos las órdenes que el DOS, con mucha autoridad, visualiza en la pantalla y dejémosle trabajar, recordando que el nuevo disco del que se habla es un disco virgen o bien uno usado pero que queremos utilizar, una vez "limpio", como si fuera completamente nuevo.

Mientras el sistema trabaja nos lo indica con un mensaje en la pantalla:

Formateando...

Cojamos la lista que hemos hecho del índice de los ficheros del disco DOS y examinémosla buscando todos los ficheros que tengan como extensión ".COM". Se trata de todos los comandos que no se cargan en memoria en el momento de la inicialización del DOS, sino solamente cuando se tiene necesidad de ellos. Dichos comandos se definen como de tipo *externo*, o no residentes, mientras que los residentes en memoria se llaman de tipo *interno* y están contenidos en el fichero COMMAND.COM (vea en el apéndice la lista de los comandos y su tipo).

En resumen, la carga del DOS equivale a la obtención inicial, a partir del disco correspondiente, de tres ficheros:

IBMBIO.COM
IBMDOS.COM
COMMAND.COM

Los dos primeros son los módulos de los que hablamos al principio, el BIOS y el BDOS, aunque llamados de forma diferente, mientras que el tercero es el procesador de los comandos. Si busca los dos primeros en la lista que obtuvo antes no los encontrará, puesto que se trata de dos ficheros "algo especiales" y el comando DIR no los toma en consideración. Están protegidos y sólo pueden ser anulados con la utilización de los comandos FORMAT o DISKCOPY. Los comandos de tipo *interno* están siempre disponibles y su ejecución es muy rápida porque no es necesario leerlos del disco.

Considerando que estamos hablando de este tema, es también oportuno indicar que no siempre el fichero COMMAND.COM está residente en memoria (Fig. 1). De hecho, existen programas de aplicación que pueden recubrir parte de la memoria que contiene el procesador de los comandos. Cuando, al final de su ejecución, se restituye el control al DOS, éste ya no está en condiciones de ejecutar comandos de gestión de los ficheros. El siste-

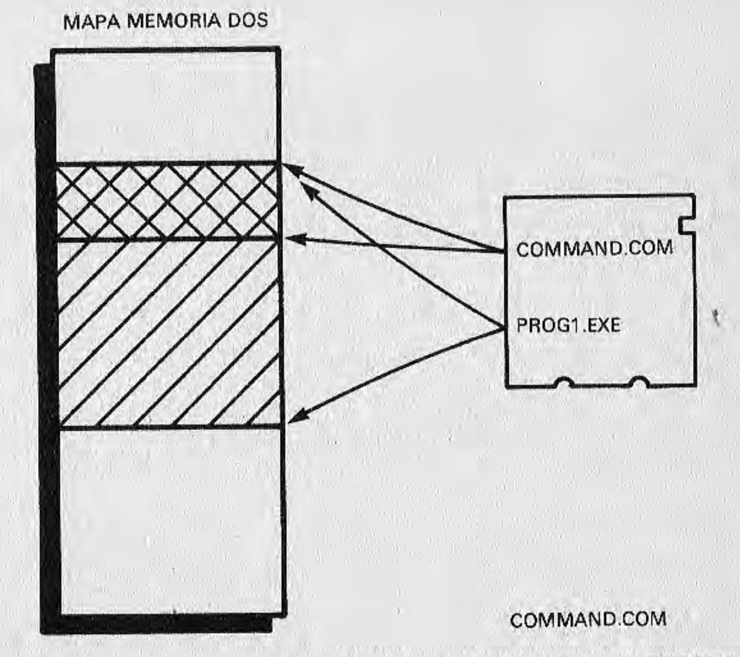


Figura 1.— La parte de memoria ocupada por el fichero COMMAND.COM (parte transitoria) está recubierta por algunos programas de aplicación. Al final de su ejecución, el DOS busca en la unidad supuesta el fichero COMMAND.COM para cargarlo de nuevo.

ma operativo busca entonces en el disco por defecto (o sea, el que aparecía en el mensaje del DOS de tipo "A>"); si lo encuentra lo carga, pero si hemos sustituido el disco y no lo localiza, el DOS nos pedirá que introduzcamos el disco con el DOS y que pulsemos cualquier tecla.

Volvamos ahora al comando FORMAT que, entretanto, hizo "algo de ruido" al trabajar durante unos 45 segundos y al escribir en la pantalla una vez terminada su tarea:

Formateando ... Formato terminado

362496 bytes totales en disco
362496 bytes disponibles en disco

Otro disco a formatear (S/N)?

Aquí tenemos que decidir si continuar el formateado de discos o si nos basta con el que acabamos de preparar.

Hemos obtenido a partir del comando FORMAT dos únicas informaciones: la relativa a los bytes que tenemos en el disco y la de los bytes que están a nuestra disposición. Si utilizamos el comando FORMAT con el parámetro S es posible transferir durante el formateado los tres ficheros que constituyen el DOS, es decir, llevar al disco también el sistema operativo de modo que no tengamos que recurrir siempre al disco del DOS. Probemos ahora a introducir el comando

A>FORMAT/S

y el sistema operativo nos proporcionará estas informaciones:

Formateando ... Formato terminado

362496 bytes totales en disco
40960 bytes utilizados por el sistema
321536 bytes disponibles en disco

Otro disco a formatear (S/N)?

Los discos preparados de este modo pueden contener las diversas aplicaciones que queramos utilizar y, al mismo tiempo, "poner en marcha" el sistema sin necesidad de emplear el disco del DOS traducido. No obstante, es necesario recordar que el comando FORMAT no transfiere los ficheros de "personalización" del teclado (KEYBSP.COM) y el fichero AUTOEXEC.BAT que son también necesarios.

Otro cometido confiado al comando FORMAT es comprobar el disco en búsqueda de pistas defectuosas que, una vez individualizadas, se marcarán para no volverse a emplear. El comando FORMAT indica las pistas defectuosas de modo que el DOS no vuelva a tomarlas en consideración. Al final de esta operación se visualizará el mensaje

xxxxxx bytes en sectores inutilizables

La preparación del disco se realiza tomando en consideración las características de la unidad en la que se efectúa, y así, para la versión 2.0, se tendrá:

— unidad de disco de una cara, los bytes son solamente 184.320;
— unidad de disco de doble cara, los bytes son 368.640.

Los discos de una sola cara pueden utilizarse en unidades de disco de doble cara, pero no es posible lo contrario.

Se puede preparar un disco de una cara también en una unidad de doble cara, proporcionando al comando FORMAT el parámetro "/1". También es posible formatear con la versión DOS 2.0 discos a emplear con la versión 1.1, mediante el parámetro "/8" (o lo que es lo mismo: 8 sectores por pista en lugar de 9).

El parámetro "/B" prepara el disco para su utilización (con cualquier versión del DOS) sin que se cargue el sistema operativo. Esta opción es de utilidad siempre que se quiera un disco destinado a contener datos, puesto que se ahorra espacio. También es posible añadir los tres ficheros fundamentales del sistema DOS con el empleo del comando SYS.

Un último parámetro utilizable es "/V", con el cual podemos etiquetar el disco con un nombre (etiqueta de volumen); este nombre se grabará en el disco y servirá para su reconocimiento.

Mencionamos anteriormente el control que el comando FORMAT tiene sobre las pistas; por consiguiente, no hemos de olvidar reformatear un minidisco si hemos descubierto que existen ficheros que ya no logramos leer (sin olvidarnos de guardar todos los demás ficheros antes de destruirlos con el nuevo comando FORMAT). Otro sistema más sencillo, pero que es válido sólo en algunas ocasiones, consiste en llamar al fichero deteriorado con un nombre particular (RENAME x, y PISTA.DIF) y, a partir de ese momento, no volver a utilizarlo.

DISKCOPY, duplicando discos

Este comando, de tipo externo, lo podemos utilizar, como ya vimos con anterioridad, para efectuar una copia de reserva de todos los discos que queramos estar seguros de no perder.

Pero, ¿por qué hacer las copias? Una respuesta podemos encontrarla leyendo el manual del MS-DOS. En dicho manual, o en los libros y en las revistas especializadas que tratan este tema, se enumeran las cosas que no se deben hacer con los discos y, por consiguiente, no las repetiremos por completo; nos limitaremos a resumir que tocar las partes no protegidas, doblar demasiado los discos, dejar caer sobre ellos polvo, ceniza de cigarrillos o líquidos, tenerlos cerca de fuentes de calor o radiación, no va nada bien a la salud de los discos flexibles.

Debemos tratar siempre de atenernos a estas reglas, pero, a veces, la prisa puede dar lugar a inconvenientes muy desagradables. Si nunca lo ha experimentado no puede imaginar todavía el descorazonamiento que sobreviene cuando se percata uno que todo un disco lleno de informaciones pacientemente recogidas está vacío, completamente vacío. Perderá mucho tiempo en tratar de recordar lo que ha sucedido antes de convencerse de que aquél era precisamente el disco en el que ha recogido todo sin hacer una copia. Empleará otro tanto de tiempo en tratar de reconstruir lo que pueda, pero no logrará casi nada. En nuestro caso el disco tenía una etiqueta PRUEBA y la única explicación posible es que se hubiera utilizado, quizá por uno mismo, para algún "condenado" experimento.

Por consiguiente, no hemos de olvidar etiquetar los discos con cuidado y protegerlos lo más posible, pero tampoco debemos olvidar que hay que dedicar un poco de tiempo a hacer copias, al menos de los que tienen más interés.

Después de lo dicho para tratar de explicar la importancia de las copias y, por lo tanto, la razón por la que debemos utilizar el comando DISKCOPY, llegamos al punto en que es preciso explicar su forma de utilización. Habida cuenta de que el disco del DOS traducido es uno de los que utilizaremos con mayor frecuencia, debemos proceder a hacer una copia de reserva del mismo.

Mientras se efectúa la copia, el comando comprueba que el disco de destino está ya formateado como el de origen o fuente. Si no está formateado o si lo está de modo diferente (versión DOS diferente, con 8 sectores en lugar de 9, etc.) procederá al formateado durante la fase de copia.

Una vez comprobado que el disco que usaremos como destino no tiene nada que nos interese, le aplicaremos una etiqueta

adhesiva con la indicación "copia DOS TRADUCIDO" y escribiremos:

```
A>DISKCOPY A: B:
```

Este comando significa que todo el contenido de A se transferirá a B. Como es habitual, seguiremos las instrucciones que el DOS nos proporcione y comenzará el trabajo.

Durante la fase de copia es posible que se detecten errores de lectura, y por ello, después de haber intentado varias veces efectuar la operación, el sistema DOS "renuncia" e indica la unidad, la pista y la cara erróneas, y continúa con su trabajo. En este caso, la copia del disco puede resultar inutilizable. El comando DISKCOPY copia todos los sectores, aunque no se empleen.

Es conveniente, una vez terminada la copia, utilizar el comando DISKCOMP que permite comprobar que la copia es idéntica al original, así como probar la copia del disco antes de archivarlo o bien archivar el original y utilizar la copia.

Hay que prestar atención al hecho de que cuando adquirimos paquetes de programas de aplicación no es frecuente que podamos copiar los datos originales porque están protegidos, pero, por lo general, en el manual de instrucciones se incluye el procedimiento que nos permite efectuar al menos una copia de reserva. En estos casos no funcionará el comando DISKCOPY.

COPY: para no perder ficheros

En Informática es sumamente importante la posibilidad de recuperar la información que, con excesiva frecuencia, por nuestras acciones indebidas o por hechos accidentales, puede desvanecerse en la nada.

Para tratar de no cometer un error que nos sería fatal, recordaremos cómo actúa el MS-DOS cuando maneja los ficheros: mantiene en memoria todas las informaciones relativas a los ficheros que estamos procesando y las transfiere al disco solamente en el momento en que le decimos que cierre dichos ficheros. Esta manera de trabajar permite al MS-DOS ahorrar una multitud de accesos al disco, pero constituye un "terrible peligro" si, durante nuestro programa, antes de haber ordenado el cierre de todos los ficheros abiertos durante el proceso, procedemos a cambiar el disco. En este caso, el DOS escribe el "viejo" directorio en el nuevo disco, destruyendo irremediablemte todas las direcciones de acceso y haciéndolo prácticamente inutilizable. Esperamos que no incurra nunca en un error tan nefasto, especialmente si no ha hecho copias del disco y del fichero (Fig. 2).

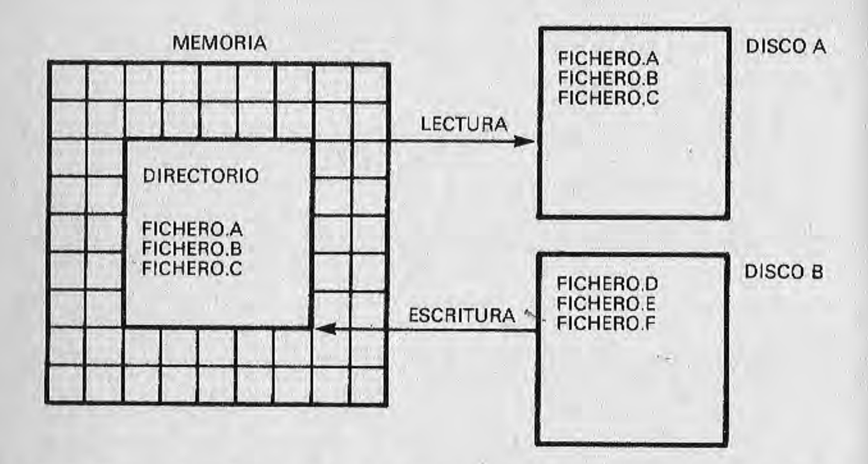


Figura 2.— Sustitución del disco durante el trabajo. Si después de haber cargado en memoria el directorio del disco A usamos uno de los ficheros existentes en el disco y después sustituimos el disco A por el B antes de haber mandado el cierre del fichero, el MS-DOS no se percatará de la sustitución. El directorio del disco A se escribirá en el disco B borrando su directorio y haciendo ilegibles todos sus ficheros.

Otros errores pueden "volatilizar" también todo nuestro trabajo; por ello hemos de aprender a hacer copias de nuestros ficheros y deberemos hacer de esta tarea un hábito.

Existen también otros motivos, además del miedo al desastre, que hacen necesario efectuar copias. Enumeremos algunos de ellos:

- Si queremos elaborar un nuevo fichero cuya estructura y, ocasionalmente, también parte del contenido forme ya parte de un fichero creado con anterioridad por otras razones (por ejemplo, un fichero que contenga un borrador de carta o una serie de subrutinas en BASIC que puedan constituir ya la base de un nuevo programa, etc.) es suficiente copiarlo asignándole otro nombre y luego modificar, actualizar, "cortar y coser" el nuevo fichero;
- necesitamos una copia del fichero para otro trabajo o aplicación o bien otro usuario la necesita; en estos casos es conveniente no fiarse y hacer una copia del original;
- queremos, por ejemplo en el tratamiento de textos, cambiar muchas cosas, pero no deseamos encontrarnos, en un determinado momento, en condición de no estar seguros de lo que estamos haciendo. Es más oportuno, en este caso, hacer una copia, trabajar con ella y si al final el resultado nos agrada, incluso podemos borrar el original;
- hemos modificado un fichero de un disco del que hicimos ya la copia de reserva. En este caso, es más rápido sustituir un fichero que hacer la copia de todos los ficheros con el comando DISKCOPY;
- otro caso en el que conviene efectuar la copia y no usar el comando DISKCOPY es cuando los ficheros de un disco están sujetos a variaciones muy frecuentes. El DOS asigna el espacio para los ficheros escribiendo en el primer sector que encuentra libre. De este modo, los ficheros frecuentemente actualizados estarán algo fragmentados, con la consecuencia de "tener siempre de paseo" a la cabeza de escritura, con las consiguientes pérdidas de tiempo. En este caso, es de gran utilidad copiar todos los ficheros de un disco en otro con el comando

A>COPY A:*.*B:

Con este comando, el DOS copia todos los ficheros uno tras otro, utilizando los sectores libres de forma secuencial y así pone todo en orden. Es, no obstante, evidente que después de este comando no es posible utilizar el DISKCOMP.

En el ejemplo que acabamos de describir encontramos de nuevo los caracteres globales que son muy útiles, pero también peligrosos, por lo que antes de emplearlos quizá sea conveniente tratar de ver qué resultados obtenemos utilizando el comando DIR con los mismos parámetros.

El comando COPY es de tipo *interno* (también se denomina "residente", puesto que reside en memoria). Tiene muchas posibilidades de utilización, algunas de ellas muy interesantes. Veámoslas:

- copiar un fichero o más en otro disco; solamente si se lo decimos proporciona un nombre diferente al fichero copiado. En este caso, la copia puede efectuarse en el mismo disco de origen;
- mientras se efectúa la copia es posible concatenar en la salida varios ficheros de entrada;
- resulta posible transferir datos desde un dispositivo del sistema a otro, por ejemplo, desde el teclado a un fichero de forma directa, o bien, desde el teclado a la impresora.

A tal fin sirven los nombres reservados de los dispositivos, que son los siguientes:

— CON: teclado o pantalla;
— PRN: impresora paralelo (primera de tres posibles);
— LPT1: impresora paralelo (sinónimo de la anterior);
— LPT2: impresora paralelo (segunda);
— LPT3: impresora paralelo (tercera);
— AUX: punto de conexión del adaptador de comunicaciones asíncronas;
— COM1: sinónimo del anterior (primero);
— COM2: segundo punto de conexión del adaptador de comunicaciones asíncronas;
— NUL: dispositivo inexistente; si se utiliza como unidad de salida simula la escritura, y si se emplea como dispositivo de entrada, simula una condición de final de fichero. Es de utilidad en las pruebas de aplicaciones o de programas en general para evitar la escritura de ficheros o de formularios que, en ese preciso momento, no sirven para nada.

Veamos ahora algunos ejemplos que, como siempre sucede, valen más que mil palabras. Comenzaremos precisamente con los nombres reservados. Probemos a dar el comando

A>COPY CON:PRN

después de haber pulsado la tecla de introducción, podemos utilizar la pantalla como si fuera una hoja de papel en blanco en la que podemos escribir lo que queramos, utilizando la tecla de introducción como si fuera la tecla de retorno del carro. Cuando consideremos que hemos acabado basta pulsar la tecla F6, que simula el final del fichero, y después de pulsar una vez más la tecla de introducción se iniciará la impresión.

Probemos ahora a introducir

A>COPY CON: FICHERO.PRU

Después de introducir una línea tras otra pulsamos la tecla F6. También en esta ocasión se inicia la escritura, pero esta vez se produce en el disco.

Veamos otros ejemplos:

A>COPY DISKCOPY.COM
A>COPY DISKCOPY.COM A: *.*
A>COPY DISKCOPY.COM B:
A>COPY DISKCOPY.COM B: DISCOPIA.COM

De los cuatro comandos sólo los dos últimos proporcionan resultados satisfactorios, porque es imposible copiar un fichero con

el mismo nombre sobre el mismo disco (la única excepción posible es cuando subdividimos el disco en varias partes con diferentes directorios, pero esta circunstancia la veremos con más detalle a continuación).

El tercer comando copia el fichero con las mismas especificaciones en el disco en la unidad B:, mientras que el cuarto copia el fichero, siempre en el disco de la unidad B:, pero cambiándole el nombre por DISCOPIA.COM. Si hubiéramos omitido la indicación de la unidad B:, el fichero se hubiera copiado en el disco de la unidad A:, que así contendría dos copias, con nombres diferentes, del mismo fichero.

Recordemos de nuevo que:

A>COPY*.*B: o también
A>COPY A:*.*B:

copia todos los ficheros del disco A: transfiriéndolos al disco B: sin modificar sus especificaciones. Lo vimos con anterioridad cuando hablamos de los discos con los ficheros demasiado fragmentados.

Veamos a continuación algún ejemplo relativo a los ficheros concatenados:

A>COPY DISKCOPY.COM+FORMAT.COM B:DOSCOM.COM
A>COPY B:DISCOPIA.COM+FORMAT.COM B:DOSCOM.COM

Con estos dos comandos obtenemos el mismo resultado; en efecto, teniendo en cuenta los ejercicios realizados con anterioridad, escribimos un fichero en el disco B: que contiene la concatenación (suma o unión) de los ficheros unidos entre sí en la expresión por el signo de la suma

A>COPY B:DISCOPIA.COM + A:FORMAT.COM

El resultado se guardará en esta ocasión en el fichero B:DISCOPIA.COM porque el comando COPY, al no encontrar especificado el fichero de destino, concatena al primero todos los ficheros sucesivamente especificados, creando un nuevo fichero igual al B:DOSCOM.COM del ejemplo anterior, pero llamado ahora B:DISCOPIA.COM.

Cuando se efectúan concatenaciones de ficheros éstos suelen ser del mismo formato (ASCII o binario); si tuviéramos que concatenar ficheros de formato diferente, lo podríamos hacer con el empleo de los parámetros "/A" para ASCII y "/B" para el formato binario.

A>COPY FILE1/A+FILE2/B+FILE3+FILE4/A PRUEBA.BBB

Del ejemplo anterior se puede deducir que el parámetro debe escribirse sin dejar ningún espacio entre la especificación del fichero y su atributo. El parámetro identifica también todos los ficheros sucesivos hasta que no se encuentre otro parámetro.

A>COPY*.COM B:TOTCOM.COM

En este caso, concatenamos todos los ficheros con la extensión ".COM" en un solo fichero que escribimos en el disco B:

Se trata de un ejemplo en donde utilizamos los caracteres globales. Podemos introducir también

A>COPY *.COM+*.EXE B:MIXTO.C&E

que contendría todos los ficheros con la extensión .COM junto con los que tienen por extensión .EXE.

Imaginemos que tiene en el disco A: una serie de ficheros denominados UNO.TIT, DOS.TIT, TRES.TIT, etc., y otra serie de ficheros denominados UNO.TXT, DOS.TXT, TRES.TXT, etc. (digamos que los primeros contienen los títulos de los capítulos uno, dos, tres, y los segundos, los textos correspondientes). Si quisiéramos ahora concatenar los títulos con los textos, podríamos utilizar:

A>COPY *.TIT+*.TXT B:*.TOT

El resultado sería una serie de ficheros UNO.TOT, DOS.TOT, TRES.TOT, etc.

Existen todavía otros casos dignos de estudio, pero son un tanto peculiares, y por ello les remitimos al manual del DOS. Por si quiere probarlos le damos algunos de ellos:

A>COPY B:*.COM B:DISKCOPY.COM

El sistema DOS responde: Cont. area dest. perdido antes de la copia, pero luego ejecuta él mismo el comando sin problema alguno. En este caso, en el que el fichero de destino existe ya, se quieren concatenar entre sí todos los ficheros que tiene la extensión ".COM". También el fichero de destino es un fichero ".COM", el sistema DOS advierte una cierta contradicción solamente cuando está preparado para copiar el fichero DISKCOPY.COM concatenándolo como fichero origen. En este punto, el fichero se podría modificar:

A>COPY B:DISKCOPY.COM B:*.COM

Con este comando, el DOS comprueba si el nombre del fichero origen es idéntico al nombre del fichero de destino. Si los nombres coinciden, el sistema DOS no efectúa la concatenación del fichero origen. Se obtiene, pues, el mismo resultado del comando anterior.

A>COPY COMMAND.COM

Cambia la fecha y la hora de grabación del fichero sustituyéndola por la fecha y la hora introducidas en el momento del encendido.

TYPE, visualizar sin editar

Se trata de un comando de tipo interno, y por ello contenido en el procesador de comandos COMMAND.COM, que sirve para visualizar en la pantalla el contenido de un fichero. Son legibles solamente los ficheros en formato de texto (ASCII), mientras que los que están en binario son casi siempre incomprensibles.

En la escritura del comando no podemos utilizar los caracteres globales y debemos escribir exactamente la especificación del fichero. Si queremos escribir en la impresora lo que se visualiza podemos, antes de introducir el comando o antes de pulsar la tecla de introducción, proceder a la pulsación de la tecla CTRL y, al mismo tiempo, de la tecla PrtSc.

Por ejemplo:

A>TYPE AUTOEXEC.BAT

Es óptimo para comprobar los ficheros de cuyo contenido no estamos seguros. Si el fichero es demasiado largo para estar contenido en su totalidad en una sola presentación visual en la pantalla, recordamos que es posible pulsar CTRL y Scroll Lock para interrumpir la ejecución del comando, o bien, CTRL y NumLock para interrumpir momentáneamente el desplazamiento hacia arriba de la pantalla ("scroll"). Se trata de un comando de utilidad incluso cuando se quiere comprobar el contenido de varios discos.

RENAME, el mismo perro con distinto collar

Este comando de tipo *interno* cambia el nombre del fichero especificado como primer parámetro por el nombre y extensión suministrado como segundo parámetro:

A>RENAME B:TOTCOM.COM TIRAR.VIA

De este modo hemos cambiado la especificación del fichero en el directorio del disco situado en la unidad B:, pero no hemos variado en absoluto nada del contenido del fichero.

Con este comando podemos utilizar los caracteres globales, pero una vez más tenemos que estar atentos porque, en ocasiones, después de haber empleado este comando con los caracteres globales no logramos comprender lo que ha sucedido y tenemos que perder una gran cantidad de tiempo en tratar de reconstruir lo que hemos cambiado.

Les recomendamos, pues, tener presente que antes de escribir el comando con caracteres globales compruebe con el comando DIR cuáles y cuántos ficheros están afectados para, si fuera preciso, corregir el "tiro".

ERASE y DEL, la peligrosa decisión

Una de las actividades que se repiten de forma periódica, y con mucha frecuencia, es la de borrar los ficheros que ya no sea preciso mantener en los discos. Esta operación es posible con el comando del tipo *interno* ERASE (también reconocido como DEL). La supresión de los ficheros antiguos permite recuperar en los discos espacio para nuevas informaciones y evita también cometer el error de utilizar ficheros no actualizados.

No obstante, hemos de tener cuidado al borrar los ficheros en no adquirir la manía de la "limpieza", porque una vez borrados dejan de existir. Si no hemos tenido la precaución de realizar una copia de reserva, nos podemos ver obligados a tener que volver a crear todo el fichero.

También en este caso vale la pena comprobar el nombre de los ficheros que pretendemos suprimir, utilizando primero el comando DIR. Hay que recordar que el sistema operativo *no* pregunta si estamos seguros de lo que queremos hacer, sino que se limita a ejecutar inmediatamente la orden que se le da con el comando.

Volvamos a nuestro disco B: Podemos, con toda tranquilidad, borrar el fichero TIRAR.VIA.

A>ERASE B:TIRAR.VIA

Considerando que pretendemos hacer un poco de "limpieza" en el disco B: cambiamos sin más la unidad asignada:

A>B:

Y el sistema DOS nos responderá con

B>

Así podemos dar ahora todas las órdenes sin repetir la unidad y sin correr el riesgo, al no haberla especificado, de borrar un fichero con el mismo nombre, pero situado en el disco A:. Demos la orden:

B>DIR

y comprobemos todos los ficheros que no nos sirven, para poder escribir correctamente el comando ERASE o DEL (es el único comando que tiene dos nombres). Es muy rápido por dos motivos: la primera razón es que se trata de un comando de tipo interno y la segunda es que solamente borra la referencia del fichero en el índice de los ficheros. Automáticamente, los sectores ocupados por el fichero se declaran libres, sin efectuar ningún otro borrado (solamente los comandos FORMAT y DISKCOPY ejecutan materialmente la escritura de todo el disco flexible).

Teniendo en cuenta los caracteres globales y considerando que el contenido del disco B: no nos interesa en absoluto, tratemos de borrar todos los ficheros de un golpe:

B>ERASE*.*

Solamente en este caso, el DOS nos pregunta:

Esta seguro (S/N)?

Respondamos con N. Escribamos después:

B>DEL

También en esta ocasión el DOS nos pregunta si estamos seguros de lo que queremos hacer. De hecho, a falta de información sobre la especificación del fichero, el DOS supondrá que es *.*

Otra peculiaridad es la de no poder borrar los ficheros IBMBIO.COM e IBMDOS.COM que están protegidos, y solamente los comandos FORMAT y DISKCOPY pueden destruirlos.

CHKDSK, verificando disquetes

Este comando permite analizar el directorio y la tabla de asignación de los ficheros de un disco y nos proporciona una presen-

tación visual en la que se resume todo lo que nos interesa saber sobre el disco en cuestión y sobre la memoria.

Se trata de un comando de tipo *externo* útil sobre todo para comprobar el estado de los discos.

A>CHKDSK B:

El comando pretende comprobar el estado del disco B. Tendremos:

Volumen DISCOPRUEBA Creado 20 OCTUBRE 1984 11:00
362496 bytes totales en disco
22528 bytes en 2 archivo(s) no visualizados
1024 bytes en 1 directorio(s)
333824 bytes en 34 archivo(s) del usuario
5120 bytes disponibles en disco
131072 memoria total en bytes
92656 bytes libres

A partir de esta serie de informaciones podemos conocer el nombre del disco, cuándo fue creado, cuántos bytes contiene, cuántos bytes ocupan los dos ficheros que no es posible visualizar (en este caso se trata de los dos módulos IBMBIO y IBMDOS), cuántos bytes hay en los subdirectorios, cuántos ficheros tenemos en el disco y cuántos bytes ocupan en total, cuántos bytes tenemos todavía disponibles y, finalmente, dos informaciones suplementarias relativas, en esta ocasión, a la memoria RAM: el total instalado y los bytes que están todavía disponibles.

Cuando enviamos el comando CHKDSK hemos de estar atentos porque no nos pide que introduzcamos el disco en el que queremos efectuar la comprobación, sino que se pone inmediatamente a trabajar en la unidad supuesta o en la especificada. Es, pues, importante que la unidad especificada sea diferente de la supuesta por omisión, a no ser que se quiera comprobar el disco que contiene el fichero CHKDSK.COM que, en este caso, es el disco con el DOS.

Este comando puede tener como parámetros incluso el nombre de uno o varios ficheros (con caracteres globales). En este caso tendremos en la pantalla, además de las informaciones ya vistas, también las relativas a la ocasional fragmentación de los ficheros. Por ejemplo, demos el nombre de un fichero al comando

B>CHKDSK A:CAPITU3.TXT

El resultado es idéntico al anterior en lo que respecta a la cantidad de bytes en el disco y en la memoria, pero leemos también estas dos líneas suplementarias:

A: CAPITU3.TXT

Contiene 6 bloques no contiguos.

Si utilizamos los caracteres globales (tipo *.*.TXT y similares), conoceremos los ficheros que contienen bloques no contiguos o errores de asignación. Tengamos presente que los ficheros con muchas áreas no contiguas producen una disminución de las prestaciones, por cuanto que el DOS ya no puede efectuar la lectura de manera secuencial.

Pueden ponerse de manifiesto también los sectores que ya no pueden leerse o que tienen algún error en su asignación. En este caso, es posible recuperar una parte del fichero especificando el parámetro "/F" acerca de cuya utilización le recomendamos la máxima prudencia y leer detenidamente el manaual para no correr el riesgo de poner en peligro la integridad del propio disco.

Si queremos obtener la indicación de cómo se realiza la comprobación o más detalles sobre los errores encontrados, podemos proporcionar el parámetro "/V".

Algunas recomendaciones

A continuación damos algunos consejos bastante importantes que estamos seguros que la mayor parte de los lectores ponen en práctica a diario, pero siempre es útil recordarlos, sobre todo para quienes no sean todavía demasiado expertos:

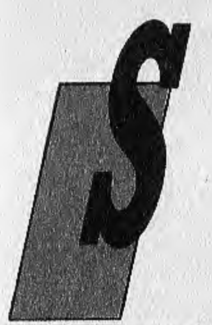
- Realicemos siempre, y hagamos de ello una costumbre, la copia de reserva;
- si los comandos no funcionan, o si no hacen lo que quisiéramos que hicieran, hemos de comprobar siempre si lo que hemos escrito es correcto, si el disco es el que pretendemos utilizar, si el fichero está contenido precisamente en ese disco o en ese directorio, si la sintaxis del comando es correcta, si se han puesto detrás de la denominación de la unidad los dos puntos (:), si la unidad especificada es la misma y si la extensión está escrita de forma correcta.
  Si después de todas las comprobaciones el comando sigue sin funcionar adecuadamente tendremos que revisar el manual y leer todo lo que concierne al propio comando;
- hemos de prestar la máxima atención a la utilización de algunos comandos, sobre todo a los que borran o modifican los ficheros. Asimismo, hemos de tener cuidado con los caracteres globales en las especificaciones de los ficheros; podemos utilizar el comando DIR para asegurarnos de que lo que pensamos coincide con lo que interpretará el MS-DOS;

- imprimamos a menudo el directorio del disco (con el comando DIR precedido por CTRL y PrtSc) e insertemos esta lista en la funda de su estuche. De este modo evitaremos perder mucho tiempo cuando tratemos de buscar ficheros que utilizamos con poca frecuencia;
- conservemos con cuidado los discos y cuando hayamos hecho manipulaciones de un fichero muy largo, hagamos una copia de dicho fichero. Obtendremos dos resultados: tener una copia de reserva y haber reorganizado el fichero escribiéndolo de forma secuencial, sin saltos;
- etiquetemos siempre los discos, busquemos un método de catalogación que nos resulte atractivo y utilicémoslo siempre, puesto que este método nos facilitará la búsqueda de las informaciones;
- clasifiquemos las copias de reserva de los discos de modo que no se pierdan, por causas accidentales, ni el original ni la copia;
- probemos primero lo que queremos realizar, y solamente cuando estemos seguros de que todo funciona adecuadamente, ejecutemos el trabajo propiamente dicho.

CAPITULO IV

EL DISCO DURO Y LOS ÍNDICES EN ÁRBOL

Formateado del disco duro



Si nuestro ordenador personal IBM tiene un disco duro (o rígido) será necesario prepararlo (formatearlo) como se hace con los disquetes (discos flexibles). Para esta preparación es preciso usar el comando FDISK, que permite preestablecer el disco duro como mejor nos convenga.

Cuando preparemos el disco, podremos "rebanarlo" según nuestros deseos sin ningún problema en absoluto y así, si utilizamos más de un sistema operativo, podemos dedicar una "rebanada" de nuestro disco al MS-DOS, otra al CP/M, otra al UCSD, etc. Esta distribución no es permanente y podremos cambiarla cuando decidamos emplear otro sistema operativo o bien no volver a utilizar uno.

Por supuesto, es posible con los comandos BACKUP y RESTORE salvaguardar toda una partición (rebanada del disco dedicada a un sistema operativo) o todas las particiones existentes para luego, después de haber reestructurado el disco con un número de particiones diferente, restablecer las particiones que queramos. El comando FDISK es fácil de emplear, habida cuenta de que todas las operaciones son guiadas por cuadros con menús muy claros y sencillos y, por consiguiente, consideramos que no vale la pena detenernos demasiado en este comando. No obstante, hay que hacer un par de consideraciones.

Es posible hacer que sea el disco duro el disco original para el MS-DOS; en este caso será necesario, cuando solicitemos el comando de FORMAT, introducir:

```
A>FORMAT C:/S/V
```

Ha de recordarse también que el formateado del disco duro exige bastantes minutos de trabajo, que contiene muchos más datos que un disco flexible, que es más rápida la lectura de los datos, que se ahorra tiempo evitando la manipulación de los discos, que es posible crear ficheros muy grandes y que, de cualquier modo, se pueden emplear los discos duros como si fueran disquetes.

Estructura en árbol de los directorios

Imaginemos ahora que comenzamos a copiar ficheros en nuestro disco y queremos comprobar, en un momento determinado, a qué punto hemos llegado o bien que queramos buscar un fichero entre los que existen, pero que no recordamos muy bien con qué nombre lo hemos escrito en nuestra unidad C: (las unidades para los discos fijos suelen ser C: y D:). Si introducimos el comando

C>DIR

veremos "desfilar" una pantalla tras otra de ficheros sin posibilidad de enteramos demasiado, a no ser que dispongamos de un índice, impreso con mucha frecuencia y por consiguiente actualizado, con los comentarios pertinentes junto a cada fichero.

Para evitar dicho inconveniente, podemos emplear el disco como si fuera la suma de muchos disquetes DISCOUNO, DISCODOS, DISCOTRES, etc., pero tampoco en este caso habremos resuelto muy bien nuestro problema. En lugar de ello pensemos en lo que queremos almacenar y escribamos una lista de todas las materias que queremos tratar:

Aplicaciones
Datos personales
Datos de la oficina
Informaciones caseras
Datos de los hijos
Listas telefónicas
Etc.

Ahora que los tenemos ante nuestros ojos podemos buscar una relación lógica entre ellos y tratar de reagruparlos en una estructura propiamente dicha. Esta operación se ilustra en la tabla siguiente. Así se tiene el árbol mostrado en la Figura 1.

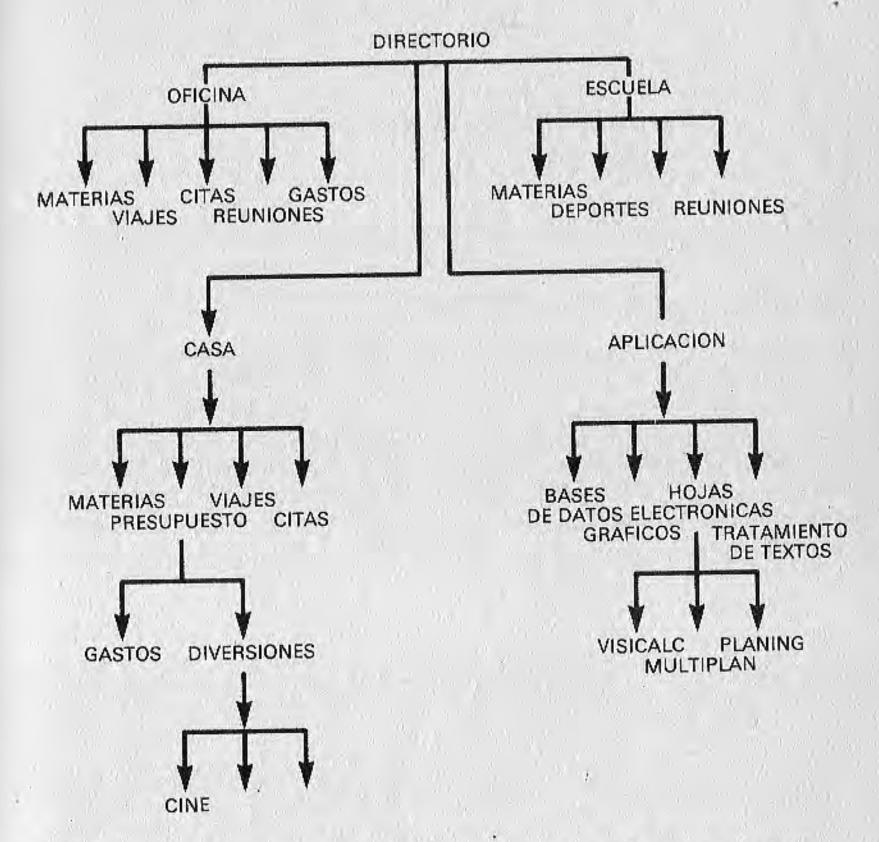


Figura 1. Estructura en árbol con directorios a diversos niveles.

Oficina	Casa	Escuela	Aplicación
Materias	Materias	Materias	Bases de datos
Viajes	Presupuesto	Deportes	Gráficos
Citas	Viajes	Deberes	Hojas electrónicas
Reuniones	Citas	Reuniones	Proceso de textos
Gastos			

Como ya dijimos, podemos considerar todos estos elementos como partes integrantes de un árbol en el que la raíz contiene los nombres de los cuatro grupos que hemos individualizado y cada uno de los cuatro grupos contiene los nombres de los argumen-

tos en que los hemos subdividido. En efecto, el MS-DOS permite definir un directorio-raíz en el cual estén contenidas todas las indicaciones de los subdirectorios que, a su vez, pueden contener otros directorios, y así sucesivamente hasta llegar a los ficheros propiamente dichos.

En la versión 1.1 del MS-DOS no era posible almacenar más de 64 ó 112 nombres de fichero (en una o dos caras), porque el índice del disco no podía contener físicamente más nombres, pero ahora, con esta organización, es posible la gestión en un disco rígido o en un disquete de centenares o millares de ficheros.

El límite de 64 ó 112 nombres sigue siendo válido, pero esta vez a cada uno de los situados en el índice raíz puede corresponder un nuevo subdirectorio que contenga el nivel inferior. Puesto que los subdirectorios son ficheros como los demás, no tendrán límites de amplitud. El límite de 64 ó 112 es vinculante solamente para los discos flexibles; para el disco duro el límite es mucho más amplio y depende del tamaño de la partición definida. Con esta nueva estructura el límite numérico de los ficheros viene determinado prácticamente por el espacio físico disponible en cada soporte magnético.

Los nombres de los ficheros y de los directorios siguen las reglas que vimos con anterioridad y, además, se tiene ahora la ventaja de que cada directorio puede contener un fichero que tenga el mismo nombre que otro fichero incluido en un directorio diferente (una ojeada a la tabla puede disipar cualquier duda al respecto).

La versión 2.0 del MS-DOS recuerda la unidad supuessta por defecto como estándar y, según el mismo principio, recuerda el directorio corriente (en uso) en cada unidad. Este directorio se denomina "directorio en uso" y es el directorio dentro del cual el MS-DOS realiza la búsqueda de los ficheros, a no ser que especifiquemos un directorio diferente en el que efectuar la búsqueda. En el momento de partida, el MS-DOS supone como directorio en uso el definido como "raíz".

Veamos cómo manejar estos directorios "en árbol" y cómo conseguir, a partir del directorio raíz, llegar al fichero que nos interesa. Volvamos, por un instante, a la tabla.

Consideremos que los nombres que hay escritos son los correspondientes a los diversos subdirectorios existentes en nuestro disco duro o disquete e imaginemos que queremos ver los ficheros contenidos en el dictorio Materias contenido en el directorio Casa. Escribiremos:

A>DIR \CASA\ MATERIAS

La barra inclinada inversa sirve para establecer los niveles sucesivos de los elementos direccionados; así el comando comunica al DOS que quiere que comience a buscar en el directorio raíz (la primera barra invertida) y que vaya luego al directorio CASA, en donde encuentra el nombre del directorio MATERIAS en el que, finalmente, encuentra los nombres de todos los ficheros. Si omitimos la primera barra y el directorio en uso no es el directorio-raíz sino, supongamos, el de ESCUELA, el comando no conseguirá encontrar los ficheros que nos interesan, porque no es correcto el *camino* (PATH) definido.

Probemos ahora a crear subdirectorios con el comando MKDIR o, de forma más abreviada, MD y con la mirada puesta en la figura 1 introduzcamos los comandos siguientes:

```
A>MD OFICINA
A>MD CASA
A>MD ESCUELA
A>MD APLICAC
A>MD \OFICINA \MATERIAS
A>MD \OFICINA \VIAJES
A>MD \OFICINA \CITAS
A>MD \OFICINA \REUNION
A>MD \OFICINA \GASTOS
A>MD \CASA \MATERIAS
A>MD \CASA \PRESUPUE
A>MD \CASA \VIAJES
A>MD \CASA \CITAS
A>MD \ESCUELA \MATERIAS
A>MD \ESCUELA \DEPORTES
A>MD \ESCUELA \DEBERES
A>MD \ESCUELA \REUNION
A>MD \APLICAC \BASEDAT
A>MD \APLICAC \GRAFICOS
A>MD \APLICAC \HOJAS
A>MD \APLICAC \TEXTOS
A>MD \CASA \PRESUPUE \GASTOS
A>MD \CASA \PRESUPUE \DIVERS
```

De este modo, hemos subdividido nuestro disco en muchas partes; el directorio raíz lo hemos dividido en los directorios OFICINA, CASA, ESCUELA y APLICAC, mientras que en el directorio OFICINA hemos creado otros subdirectorios (MATERIAS, VIAJES, CITAS, REUNION y GASTOS).

Pero no hemos acabado aquí, puesto que al mismo nivel hemos creado también otros subdirectorios utilizando los mismos nombres que, precisamente porque son "hojas" de otra "rama", pueden coexistir. Operando de este modo podemos crear cualquier estructura que se nos ocurra tener.

Probemos a solicitar ahora el índice de los ficheros del directorio-raíz:

A>DIR

El volumen en la unidad A es PRUEBACUADER
Directorio de A:

```
OFICINA              <DIR>            28-11-85     2:36p
CASA                 <DIR>            28-11-85     2:36p
ESCUELA              <DIR>            28-11-85     2:36p
APLICAC              <DIR>            28-11-85     2:36p
TEXTO        TXT                1933  28-11-85     3:05p
PROG1        DOC                2860   6-11-84     1:48p
DISCO        RAD                   0  28-11-85     2:54p
EDIT         EXE               58112  11-10-84    12:00p
TXT          PRF                 313   28-4-85     2:47p
PRINT        COM                4267  31-10-84    12:00p
         10 Archivo(s) 251904 bytes libres
```

y con:

A>DIR \OFICINA

El volumen en la unidad A es PRUEBACUADER
Directorio de A:\ OFICINA

```
<DIR>  28-11-85  2:49p
<DIR>  28-11-85  2:49p
<DIR>  28-11-85  2:50p

MATERIAS
7 Ficheros 251904 bytes libres

VIAJES    <DIR>  28-11-85  2:51p
CITAS     <DIR>  28-11-85  2:51p
REUNION   <DIR>  28-11-85  2:52p
GASTOS    <DIR>  28-11-85  2:53p
```

Todos los directorios, con la excepción del directorio-raíz definen dos ficheros: el primero, precedido por un punto (.), identifica el directorio como subdirectorio, mientras que el segundo, precedido por dos puntos (..), permite al MS-DOS reconocer el "padre" de este último de modo que se pueda "remontar" la eventual cadena especificando solamente los dos puntos para luego volver a descender hasta encontrar el directorio que nos interesa.

Un ejemplo siempre será ilustrativo. Teniendo siempre presente la figura 1 imaginemos que el directorio en uso es GASTOS en el ámbito del directorio PRESUPUESTO de nuestra CASA y que queremos llegar al fichero CINE que se encuentra en DIVERSIONES. Lo que debemos especificar es el *camino* que debe seguir el DOS para llegar adonde queremos:

A>TYPE A: \CASA \PRESUPUE \DIVERS \CINE

o bien, podemos escribir:

A>TYPE .. \DIVERS \CINE

En el primer caso se suministra el nombre del camino ("pathname") completo, desde la raíz a la "hoja" (fichero). En el segundo caso, el par de puntos hace que nos remontemos al directorio PRESUPUE y desde allí, en sentido descendente, llegaremos al fichero CINE.

Se observará también que, después de la ejecución del comando, se queda en el subdirectorio GASTOS.

A partir de estos ejemplos se deduce que el nombre del fichero aparece al final del comando, mientras que la especificación de la unidad aparece al comienzo. La secuencia más larga de camino, o de recorrido, no puede ser superior a los 63 caracteres. Para copiar ficheros en los diversos subdirectorios se utiliza el comando COPY con la especificación del camino y del nombre del fichero. Probemos a introducir el disco en la unidad B:

A>COPY B:ARBOL.COM \APLICAC \HOJAS

Así hemos transferido el fichero ARBOL.COM al subdirectorio \APLICAC \HOJAS y, del mismo modo, podemos transferir todos los ficheros que nos interesen.

Antes de proseguir, solicitemos la lista de algunos subdirectorios:

A>DIR \CASA \PRESUPUE \DIVERS

El volumen en la unidad A es PRUEBACUADER
Directorio de A: \CASA \PRESUPUE \DIVERS

```
                                28-11-85     2:42p
                                28-11-85     2:42p
                  <DIR>  1203   28-11-85     2:48p
                  <DIR>
CINE         3 Ficheros 250880 bytes libres
```

y también:

A>DIR \APLICAC \HOJAS

El volumen en la unidad A es PRUEBACUADER
Directorio de A: \APLICACION \HOJAS

.	DIR			28-11-85	2:41p
..	DIR			28-11-85	2:41p
VISICALC			2860	4-6-84	1:48p
MULTIPLA			2860	4-6-84	1:48p
LOTUS123			2860	4-6-84	1:48p
ASSISTAN			2860	4-6-84	1:48p
ARBOL	COM		1572	31-10-83	12:00p
PRINT	COM		4827	31-10-83	12:00p

8 Ficheros 244736 bytes libres

Comentemos otros comandos para el control de los subdirectorios:

- RMDIR(RD) - Borra el subdirectorio, pero solamente si este último ya no contiene ningún fichero ni otro subdirectorio; tiene efecto sobre un solo directorio cada vez. Si se utiliza el camino se borra el último nombre que aparece en dicho recorrido. No es posible borrar el directorio-raíz ni el directorio en uso.
- CHDIR(CD) - Sirve para indicar al MS-DOS qué directorio utilizar para la búsqueda de los ficheros a partir del momento en que se ejecuta, es decir, cambia el directorio en uso. Puede especificarse un camino o, para el directorio-raíz, la barra invertida solamente (CD \).
- TREE - Genera una presentación visual que describe la estructura completa de los directorios contenidos en un disco. En nuestro caso sería:

```
LISTA CAMINO DIRECTORIOS
DEL VOLUMEN PRUEBACUADER

Camino: \OFICINA
Subdirectorios: MATERIAS
                VIAJES
                CITAS
                REUNION
                GASTOS
Camino: \OFICINA \MATERIAS
Subdirectorios: ninguno
```



```
Camino: \OFICINA \VIAJES
Subdirectorios: ninguno
Camino: \OFICINA \CITA
Subdirectorios: ninguno
Camino: \OFICINA \REUNION
Subdirectorio: ninguno
Camino: \OFICINA \GASTOS
Subdirectorio: ninguno
Camino: \CASA
Subdirectorio: MATERIAS
               PRESUPUE
               VIAJE
               CITAS
Camino: \CASA \MATERIAS
Subdirectorios: ninguno
Camino: \CASA \PRESUPUE
Subdirectorios: GASTOS
                DIVERS
Camino: \CASA \PRESUPUE \GASTOS
Subdirectorios: ninguno
Camino: \CASA \PRESUPUE \DIVERS
Subdirectorios: ninguno
Camino: \CASA \VIAJE
Subdirectorios: ninguno
Camino: \CASA \CITAS
Subdirectorios: ninguno
Camino: \ESCUELA
Subdirectorios: MATERIAS
                DEBERES
                DEPORTES
                REUNION
Camino: \ESCUELA \MATERIAS
Subdirectorios: ninguno
Camino: \ESCUELA \DEPORTES
Subdirectorios: ninguno
Camino: \ESCUELA \DEBERES
Subdirectorios: ninguno
Camino: \ESCUELA \REUNION
Subdirectorios: ninguno
Camino: \APLICAC
Subdirectorios: BASEDAT
                GRAFICOS
                HOJAS
                TEXTOS
Camino: \APLICAC \BASEDAT
Subdirectorios: ninguno
```

Camino: APLICAC \GRAFICO
Subdirectorios: ninguno
Camino: APLICAC \HOJAS
Subdirectorios: ninguno
Camino: APLICAC \TEXTOS
Subdirectorios: ninguno

- PATH - Permite al MS-DOS ir a buscar los comandos y los ficheros de tipo "batch" (BAT) —por lotes— que no estén contenidos en el directorio en uso, especificando el camino a recorrer.

  Los puntos del camino deben estar separados por una barra inclinada invertida y han de estar especificados todos ellos, pues el MS-DOS no tiene en cuenta, en la búsqueda, los caminos supuestos por omisión para llegar al directorio en uso. También es posible especificar solamente la unidad y, en este caso, el DOS efectúa la búsqueda en el directorio en uso para dicha unidad.

  Probemos a ver cómo se consigue la lista de los subdirectorios que acabamos de observar considerando que el comando TREE no está en el directorio-raíz.

En efecto, si probamos a escribir

A>TREE

no obtendremos otra cosa que un mensaje de error.

Proporcionamos ahora al DOS el camino a través del cual podrá llegar a encontrar el comando, la serie de comandos o los ficheros de una aplicación, según las exigencias.

A>PATH \APLICAC \HOJAS

Volvamos a probar ahora el comando TREE y veremos aparecer en la pantalla la lista deseada.

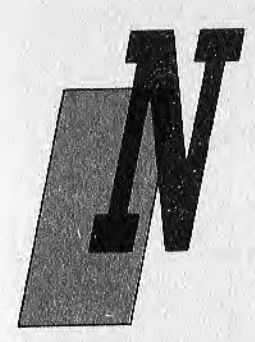
El comando PATH queda permanentemente hasta que no intervenga otro comando PATH que anule el anterior. Para volver al valor nulo es suficiente introducir

A>PATH;

Otra característica de este comando es que cualquier error no se encuentra hasta que el DOS procede a buscar de forma efectiva los comandos o los ficheros por lotes ("batch") a través del camino indicado.

CAPITULO V

PRINCIPALES COMANDOS DEL MS-DOS



No es posible, en una única monografía, hablar de todos los comandos que pueden ser utilizados por el MS-DOS exponiendo todas sus peculiaridades, pero sí es útil tener un conocimiento de todos ellos. Damos, pues, a continuación una breve descripción de los comandos y "subcomandos", junto con una indicación de sus funciones.

Comandos

- ASSIGN - asigna x=y

  Comunica al MS-DOS que la unidad "x" está asignada como unidad "y"; por lo tanto, todas las peticiones a enviar a la unidad "x" serán enviadas, desde este momento, a la unidad "y".

  Se trata de un comando de tipo externo.

  Es posible efectuar más de una asignación repitiendo el comando, pero para restablecer las asignaciones estándar es suficiente proporcionar sólo una vez el comando sin ningún parámetro.

  Es de utilidad cuando las aplicaciones no son tan flexibles que permitan la asignación de unidades diferentes a las preestablecidas. Por ejemplo, hay aplicaciones que, para funcionar, exigen la presencia de uno de los discos en la unidad A: No obstante, suele resultar cómodo cargar todos los programas en el disco rígido y evitar así la manipula-

ción de los discos flexibles y, en este caso, es preciso efectuar la asignación de la unidad A como unidad C.

- BACKUP - la seguridad

Es el comando que permite realizar copias de reserva de los ficheros existentes en disco duro, tanto en el directorio-raíz como en todos los subdirectorios.

Se trata de un comando de tipo externo y puede utilizarse con el nombre propio del fichero o bien con los caracteres globales, especificando un camino o proporcionando parámetros especiales. Estos últimos permiten efectuar la conservación de todos los ficheros especificados (/S) incluyendo todos los de los subdirectorios, solamente los ficheros actualizados después de una determinada fecha (/D), sólo los ficheros que se han modificado después de la última copia de reserva ("backup") efectuada (/M) o bien permiten añadir ficheros en un disco que ya contiene ficheros salvaguardados (/A). Si no se utiliza este último parámetro, el MS-DOS solicita siempre un nuevo disco para grabar los ficheros, destruyendo todos los que existieran en él antes, tal como sucede con los comandos DISKCOPY y FORMAT.

Por ejemplo:

A>BACKUP C:*.TXT A:

De este modo se efectúa la copia de reserva de todos los ficheros del disco C que tengan la extensión .TXT. Si un disco no es suficiente, el DOS solicitará que se suministre un segundo, y así sucesivamente hasta finalizar la copia. Será necesario, por supuesto, numerar los discos en el orden requerido para poder efectuar una eventual restauración (RESTORE).

El ejemplo siguiente efectúa la grabación de todos los ficheros del disco rígido C; la barra invertida indica el directorio-raíz y el parámetro "/S" todos los subdirectorios.

A>BACKUP C:\A:/S

Ahora imaginemos que se tiene la estructura en árbol descrita en la figura 1 del capítulo 4 en el disco rígido C y que se quiere realizar una copia de reserva del fichero CINE y solamente de dicho fichero. Para ello tecleamos:

A>BACKUP C:\CASA\PRESUPUE\DIVERS\CINE A/A

En este caso, la copia se efectúa en el disco ya introducido en la unidad A; si no especificamos el parámetro "/A", el DOS nos pediría que introdujéramos un nuevo disco.

- BREAK - paradas a la carta

Se trata de un comando de tipo interno y funciona como un interruptor que controla la petición de interrupción efectuada desde el teclado introduciendo el comando Ctr-Break (pulsación simultánea de las dos teclas). Prácticamente permite la interrupción.

En condiciones normales, el MS-DOS comprueba la existencia de la petición sólo durante la ejecución de operaciones con la pantalla, el teclado, la impresora y todos los dispositivos auxiliares. Cuando se introduce el comando BREAK ON, el MS-DOS comprueba la existencia de la petición sea cual sea la operación que se esté efectuando. Es así posible interrumpir los programas que ejecutan muy pocas operaciones en pantalla o teclado, como es el caso de los compiladores.

- CHDIR (CD) - cambio de directorio

Es un comando de tipo interno que efectúa el cambio del directorio en uso en una unidad especificada.

Es necesario cuando se pretende trabajar con ficheros que están incluidos en un directorio diferente a aquel con el que se está operando. Si se utiliza sin parámetros, o sólo con el parámetro que especifica la unidad, se visualiza el camino del directorio de la unidad supuesta por defecto o de la unidad especificada.

— A>CHDIR\

vuelve al directorio-raíz de la unidad supuesta A:

— A>CD C:\OFICINA\MATERIAS

pasa al directorio de la unidad C: permitiendo comprobar los ficheros del subdirectorio MATERIAS.

Una pequeña peculiaridad, a veces muy importante, es que la búsqueda del nuevo directorio se inicia desde el directorio en uso y, por lo tanto, si se quiere descender de nivel es posible especificar solamente los subdirectorios de nivel inferior, mientras que si se desea elegir un directorio de nivel superior o en una "rama" diferente, será necesario especificar todo el camino desde el directorio-raíz (vea también el capítulo 4 en el apartado relativo a la utilización de las estructuras en árbol).

- CHKDSK - verificar el disco

  Analiza los directorios y la tabla de la asignación de ficheros proporcionando un resumen del estado del minidisco o disco indicado y de la memoria. Es de tipo externo y, si se utiliza especificando un fichero o empleando los caracteres globales, proporciona la fragmentación de los ficheros.

  Para más detalles puede referirse al capítulo 3.

- CLS - borrar la pantalla

  Este comando permite borrar completamente el contenido de la pantalla. Es de tipo interno.

- COMP - ¿son verdaderamente iguales?

  Durante la ejecución de este comando se efectúa la comparación de dos series de ficheros y, por supuesto, es posible especificar como parámetros tanto un par de ficheros como, utilizando los caracteres globales, más de un fichero, tanto en el mismo directorio como en directorios diferentes.

  Se trata de un comando de tipo externo que puede ser introducido también sin parámetros: después de haber leído el comando, el MS-DOS se "para" y nos pide que introduzcamos los parámetros. En este caso, podemos comparar también ficheros que no residan en el disco que contiene el comando.

  Asimismo se pueden comparar todos los ficheros de un directorio con los de un segundo directorio especificando solamente el camino. Mientras se está realizando la comparación se visualizan los nombres de los ficheros objeto de examen y los caminos correspondientes.

  En el caso de un error, el mensaje que se escribe en la pantalla indica la posición en la que se obtuvo la diferencia y los dos caracteres que están contenidos en los ficheros primero y segundo. El contenido y la posición en la que está el error se escriben en caracteres hexadecimales. Después de haber encontrado una diferencia, el MS-DOS prosigue con la comparación de los ficheros hasta encontrar un máximo de diez errores. Después del décimo error se "desanima", supone que ya no hay posibilidad de encontrar una igualdad y termina con el mensaje:

  10 desigualdades - final de la comparación

  Si todo resultó correcto aparecerá, por el contrario, el mensaje:

  La comparación de los ficheros fue bien

  La comparación continúa con la siguiente pareja de ficheros, pero si la serie de ficheros se termina veremos aparecer el mensaje:

  Quiere comparar otros ficheros (S/N)?-

  Llegados a este punto se procede como para los demás comandos dando una respuesta "N" para terminar o bien "S" si queremos efectuar otras comparaciones entre ficheros. En algunos casos podremos ver aparecer en la pantalla el mensaje siguiente:

  Señal de final de fichero no encontrada

  En estos casos estamos en presencia de ficheros creados por aplicaciones o programas que registran en el directorio la longitud de los ficheros en múltiplos de 128 caracteres y no la longitud real en bytes. El comando COMP no puede comprobar que el fichero termina en algún carácter antes de la última posición porque no llega a encontrar el carácter de final de fichero (Ctrl-Z).

- COPY - utilizarlo con cuidado no hace daño

  Se trata de un comando de tipo interno muy importante y muy empleado en todas sus posibles aplicaciones.

  Copia uno o varios ficheros en otro disco, en el mismo disco pero en distinto directorio o en el mismo directorio pero con distinto nombre. Permite también combinar varios ficheros entre sí, concatenándolos, y la introducción directa de información desde el teclado en un fichero.

  Más información sobre esta materia puede encontrarse en el capítulo 3.

- DATE - la fecha

  Este comando permite introducir o actualizar la fecha que luego se utiliza, por ejemplo cuando se registran los ficheros en el directorio. El formato de la fecha es el típico de los países de lengua inglesa, es decir, primero el mes y luego el día y el año.

  El programa WTDATIM puede utilizarse para poder obtener la fecha en el formato

  dd-mm-aa

Cuando se utiliza el comando, que es de tipo interno, el sistema visualiza la fecha que ha almacenado en el momento del encendido y luego nos pide que introduzcamos la nueva fecha. Si pulsamos la tecla de introducción sin escribir nada, dicha operación servirá para confirmar la fecha visualizada anteriormente.

- DEL - con mucho cuidado

  Este comando es sinónimo del comando ERASE, del que hemos hablado en el capítulo 3.

- DIR - el índice

  También de este comando hemos hablado en el capítulo 3. Se trata de un comando de tipo interno que elige en una presentación visual todos los elementos incluidos en los directorios o solamente los correspondientes a los ficheros que nos interesan; además, indica de qué volumen se trata y la cantidad de espacio libre que queda en el disco.

  En las líneas correspondientes a los ficheros indica el nombre del fichero, su extensión, la dimensión en caracteres y la fecha de la última actualización. Es posible proporcionar dos parámetros al comando: el primero, "/P", permite detener el desplazamiento ascendente ("scroll") de la presentación visual en la pantalla cuando una página está completa, y el segundo, "/W", visualiza el directorio con los nombres de los ficheros, cinco en cada línea.

- DISKCOMP - compara el contenido de dos unidades

  Efectúa una comparación entre la primera unidad especificada (x) y la segunda (y), verificando que el contenido de los discos sea exactamente igual. Se utiliza casi siempre después de un DISKCOPY y se trata de un comando de tipo externo.

  Todos los parámetros son opcionales y, por consiguiente, el comando puede introducirse sin ningún parámetro. Ello significaría que queremos comprobar dos discos que introducimos, de forma alternativa, en la misma unidad supuesta por el DOS en ese momento.

  El parámetro "/1" permite comparar solamente la primera cara de los discos, aunque sean de doble cara.

  El segundo parámetro, "/8", efectúa la comparación de los discos solamente en 8 sectores, aunque el primero de los discos se haya formateado con 9 sectores por pista.

  En el caso de que se encontraran discordancias, se visualizarían mensajes de error que indicarían la pista y la cara que contienen la diferencia.

  Al final se tendría el mensaje habitual:

  Quiere comparar otros discos (S/N)?

  Hay que tener presente que no visualiza ningún mensaje de confirmación de que los dos discos sean iguales.

  No se puede efectuar una comparación con el comando DISKCOMP si se han empleado comandos COPY, porque este comando registra el nuevo fichero de modo casi siempre diferente del original. Por consiguiente, es oportuno en este caso utilizar el comando COMP.

- DISKCOPY - copia de discos

  También este comando se ha descrito ampliamente en el capítulo 3 y, por lo tanto, sólo nos queda recordar que se trata de un comando de tipo externo y que los parámetros pueden omitirse.

  Permite efectuar una copia "a carbón" de un disco. El parámetro "/1" permite copiar solamente una cara aunque la unidad y (o) el disco sea de doble cara.

- ERASE - el "asesino"

  Permite borrar un fichero o una serie de ficheros del directorio recorrido. Se trata de un comando de tipo interno. Equivale a DEL (ver capítulo 3).

- FORMAT - da forma al disco

  Es un comando de tipo externo con muchos parámetros posibles, todos ellos opcionales.

  Su función es la de preparar los discos, flexibles o rígidos, para que puedan recibir las informaciones y grabarlas de manera organizada y recuperable.

  Los parámetros utilizables son:

  /1 - permite formatear un disco en una sola cara;
  /8 - prepara un disco con solamente 8 sectores por pista; en la práctica, define 9 sectores, pero indica al DOS que utilice solamente 8;
  /V - el comando solicita etiquetar el disco con una serie de caracteres (máximo de 11);
  /B - prepara el disco para recibir los módulos IBMBIO.COM y IBMDOS.COM, pero no los carga para que se tenga la posibilidad de elegir entre las diversas versiones disponibles (1.0, 1.1 ó 2.0) con el comando SYS.

/S - como el parámetro anterior, prepara el disco para recibir el sistema operativo, pero no se detiene en la preparación y continúa con la carga de los dos módulos y del fichero COMMAND.COM.

Para más detalles sobre este comando le remitimos al capítulo 3.

- GRAFTABL - tabla de los caracteres gráficos

Se trata de un comando de tipo externo que carga en memoria una tabla de caracteres adicionales. Estos caracteres se utilizan por el adaptador de color/gráficos para visualizar los caracteres ASCII desde 128 a 255 cuando se emplea el ordenador en el modo de gráficos. La utilización de este comando aumenta la amplitud del MS-DOS residente en 1.392 bytes.

El comando debe darse una sola vez puesto que, de no ser así, cada vez se cargará nuevamente la tabla ocupando otros 1.392 bytes.

- GRAPHICS - prepara la impresora para los gráficos

Se trata de un comando de tipo externo que permite imprimir los gráficos o las figuras que se visualizan en la pantalla en una impresora matricial para gráficos.

También este comando debe ejecutarse antes de que sea necesario y aumentará el MS-DOS residente en 688 bytes.

Para imprimir los resultados obtenidos en la pantalla es necesario pulsar simultáneamente las teclas Shift (Intercambio de Mayúsculas/Minúsculas) y PrtSc. El tiempo necesario varía desde 30 segundos a 3 minutos, dependiendo del modo en el que se utilice la pantalla en ese momento (texto, gráfico de colores en baja resolución o en alta resolución).

- KEYBxx - utiliza el teclado xx

No se trata de un comando propiamente dicho, sino más bien de una serie de programas que tienen la función de cargar en la memoria residente el programa de teclado que sustituye al ya existente en ROMBIOS (teclado americano). Es un comando de tipo externo y los programas de las versiones DOS 1.x no son compatibles en absoluto con la versión 2.0 ó 2.1.

El programa cargado cuando se inicializa el MS-DOS traducido se ejecuta cada vez que se reinicializa el procesador y, desde ese momento, el teclado se "personaliza", aunque no se borrará el programa del teclado original. Asimismo, podemos cargar otros programas de teclado, pero el primero no se borrará y quedará residente; tan sólo el control se pasará al nuevo programa de teclado.

Es posible pasar al teclado americano en cualquier momento pulsando las teclas Ctrl, Alt y F1, mientras que para volver al teclado anteriormente utilizado basta pulsar Ctrl, Alt y F2.

Los programas de teclado disponibles son cinco, además del residente, y cada uno tiene un lenguaje diferente.

— KEYBUK - Gran Bretaña
— KEYBGR - Alemania
— KEYBRF - Francia
— KEYBIT - Italia
— KEYBSP - España

- MKDIR (MD) - crea directorios

Es el comando que permite crear subdirectorios en el disco duro/flexible que se especifica como parámetro. Puede utilizarse también la forma abreviada MD. Se trata de un comando de tipo interno.

Si no se especifica la unidad, se considerará la unidad supuesta por defecto. Para más detalle sobre este comando le remitimos al capítulo 4, a los apartados relativos a la unidad de disco rígido y a la estructura de los índices en árbol.

- MODE - cambia el modo de funcionamiento

Controla el modo de funcionamiento de la impresora o de la unidad de presentación visual en conexión con el adaptador de control color/gráficos.

Establece también las diversas opciones para el adaptador de comunicaciones asíncronas y permite enviar los datos destinados a la impresora al propio adaptador.

Se trata de un comando de tipo externo que merecería un capítulo completo, pero nos contentaremos con dar un resumen de sus características.

— Control de la impresora

Con MODE se puede controlar la impresora. Para adquirir un poco de práctica comenzaremos a utilizar los diversos comandos:

MODE LPT1	elegimos la impresora número 1
MODE LPT3	elegimos la impresora número 3

MODE LPT1:132 elegimos la impresora número 1 y queremos imprimir en el modo comprimido.

MODE LPT2:80 elegimos la impresora número 2, y queremos imprimir en el modo supuesto por defecto en el momento del encendido, es decir, 80 caracteres por línea

MODE LPT2:,8 elegimos la impresora número 2; no cambiamos el número de caracteres que queremos imprimir, pero deseamos imprimir 8 líneas por pulgada y no 6 como se establece en el momento del encendido. Para volver al valor anterior es suficiente dar la orden o comando, sustituyendo el número 8 por el 6.

Existe también otro parámetro que es posible especificar y que se utiliza para repetir las tentativas de impresión en caso de error de superación del tiempo. Por ejemplo:

MODE LPT:,,p

Si se quiere interrumpir las tentativas se puede hacer utilizando las teclas Ctrl-Break.

— Control de la presentación visual

Veamos cómo es posible controlar la presentación visual. Para ello utilizaremos los parámetros que se indican a continuación:

40	- Sirve para establecer una presentación visual con solamente 40 caracteres cuando el control de la pantalla se transfiere al adaptador de color/gráficos.
80	- Sirve para establecer la presentación visual de 80 caracteres cuando el control de la pantalla se transfiere al adaptador de color/gráficos.
BW40	- Establece la anchura de la presentación visual en los habituales 40 caracteres por línea y asigna el control al adaptador de color/gráficos, pero inhibe el color.
BW80	- Igual al anterior, pero establece la anchura de la presentación visual en 80 caracteres por línea.
CO40	- El control se transfiere al adaptador de color/gráficos, habilita el color y establece la línea de 40 caracteres.
CO80	- Como el anterior, pero con la línea de 80 caracteres por línea.
MONO	- Quita el control al adaptador activo en ese momento y lo asigna al adaptador monocromático con 80 caracteres por línea.

Hay otros dos parámetros que siguen al de la especificación del adaptador y son los relativos al centrado de la presentación visual; pueden producir un desplazamiento a la derecha o a la izquierda de los caracteres visualizados para una mejor legibilidad.

MODE 80,L,T

Con este comando se establece en 80 el número de caracteres y se desplaza todo a la izquierda dos caracteres; además, el sistema pregunta si la presentación visual está como queríamos: si respondemos S se produce una parada y si respondemos N se repetirá el desplazamiento y también la solicitud, hasta que proporcionemos la respuesta S. En el caso de que quisiéramos desplazar a la derecha la presentación visual en pantalla sería necesario sustituir L por R. En caso de establecimiento de los 40 caracteres, el desplazamiento se reduce a un carácter cada vez.

— Control del adaptador de comunicaciones asíncronas

Se pueden tener dos adaptadores diferentes (números 1 y 2) con diferentes velocidades de transmisión: 110, 150, 300, 600, 1200, 2400, 4800 y 9600 baudios. El comando tiene la forma

MODE COM1:12,N,8,1,P

El primer número indica el adaptador;

el segundo, después de los dos puntos, la velocidad;

N quiere decir ningún control de paridad; O sería paridad impar, y E, par. El valor supuesto por defecto es E;

8 son los bits que componen un dato (el valor por defecto es 7);

1 significa los bits de parada (por defecto 1 bit de parada).

y nos volvemos a encontrar el parámetro P, con el mismo significado: se repiten las tentativas de imprimir en una impresora serie. Para interrumpir las tentativas es suficiente pulsar las teclas Ctrl-Break, pero para hacer cesar por completo los intentos será necesario repetir la inicialización.

— Redireccionamiento de los datos

Es posible redireccionar los datos destinados a una impresora en paralelo hacia el adaptador de comunicaciones asíncronas. Naturalmente es necesario, antes de enviar datos al adaptador, inicializarlo como hemos visto anteriormente.

MODE LPT1:=COM1 o bien MODE LPT2:=COM2

Para inhibir el comando es suficiente utilizar

MODE LPT1:80,6

Es decir, basta reinicializar la impresora.

- PATH - el camino a seguir

Se dijo anteriormente que era posible subdividir el disco en particiones diferentes, en cada una de las cuales se pueden escribir ficheros (con el comando MKDIR). También se dijo que cuando se escribe un comando o el nombe de un fichero de tipo "Batch" (.BAT), el MS-DOS efectúa la búsqueda en el directorio en uso y si no lo encuentra genera un mensaje de error. Entonces se nos plantea la interrogante de cómo se puede encontrar un comando que reside en otro subdirectorio.

En este punto entra en juego el comando PATH, que significa literalmente sendero o camino y cuyo objeto es precisamente el crear un recorrido a lo largo del cual el MS-DOS efectúe sus búsquedas. De este modo ya no se detendrá en el directorio en uso, sino que sigue las indicaciones que le son proporcionadas por el comando PATH.

Se trata de un comando de tipo interno y, por consiguiente, siempre está disponible a condición de que esté residente el fichero COMMAND.COM.

Si se tuvieran varias unidades de disco sería posible especificar también la unidad elegida para la búsqueda.

Para simplificar, imaginemos que se tiene el disco del sistema en la unidad B: y un disco de aplicación o de trabajo en la unidad A: y que utilizamos al 90 por 100 los comandos existentes en el disco A:

En estas condiciones, cuando queramos utilizar un comando del MS-DOS tendremos que escribir antes del comando la unidad en la que reside; por ejemplo:

A>B:COMP FILE1 FILE.BAK

Si, por el contrario, apenas cargado el sistema le proporcionamos el comando

A>PATH B:\

y luego cambiamos la unidad e introducimos el disco de trabajo en la unidad A:, podemos proporcionar cualquier comando sin especificar la unidad B:. Esto es así porque el MS-DOS se ha informado de que para ejecutar un comando debe efectuar la búsqueda previa en el directorio en uso de la unidad supuesta por defecto y, luego, en el directorio-raíz de la unidad B:.

B: indica la unidad en la que se realiza la primera búsqueda, después del directorio en uso, y la barra invertida sirve para indicar el directorio-raíz.

Veamos otro ejemplo de utilización del comando PATH en el que la unidad supuesta por defecto sea la A: y los posibles subdirectorios en los que efectuar la búsqueda sean:

A:\PRIMERO
A:\PRIMERO\SEGUNDO
B:\MIOCOM

El comando se escribe como sigue (observe el punto y coma utilizado como separador)

A>PATH\PRIMERO;\PRIMERO\SEGUNDO;B:\MIOCOM

Desde ese momento en adelante cualquier comando, proporcionado sin indicación del directorio en donde se encuentra, se busca primero en el directorio en uso, después en el subdirectorio PRIMERO, contenido en el directorio-raíz A: (que no se dijo que fuera el directorio en uso), después en el subdirectorio SEGUNDO y, finalmente, en el subdirectorio MIOCOM contenido en el directorio-raíz de la unidad B:

Solamente si el comando no se encuentra en ninguno de los subdirectorios especificados en el camino, el MS-DOS emitirá el mensaje

Comando o nombre de fichero no válido

En caso de que quisiéramos saber qué camino está definido se puede visualizar introduciendo el comando PATH. El DOS responde (para el ejemplo anterior):

PATH=\PRIMERO;\PRIMERO\SEGUNDO;B:\MIOCOM

Si, en cambio, queremos anular el camino en uso introduciremos el comando seguido por el punto y coma (PATH;). En este caso, el MS-DOS no mantiene ningún camino de búsqueda y considera solamente el directorio en uso de la unidad especificiada o supuesta por defecto.

Una pequeña advertencia: los caminos deben estar todos ellos bien especificados; si cometemos errores en la definición de unidades o caminos, el MS-DOS nos avisará solamente cuando se disponga a recorrer el camino "falso" que le hemos indicado.

- PRINT - la cola de impresión

Este es uno de los comandos más innovadores de la versión 2.0. Sirve para introducir el concepto de "multitarea" o, lo que es lo mismo, de trabajo simultáneo.

En efecto, este comando sirve para mandar a la cola de impresión uno o más ficheros sin "comprometer" por completo al procesador. Se pueden poner en cola de impresión hasta diez ficheros; la primera vez que se utiliza el comando, se incrementa la amplitud de memoria del MS-DOS residente en unos 3.200 bytes.

El comando es de tipo externo y los nombres de los ficheros deben escribirse uno tras otro separados por un espacio en blanco y precedidos, si fuera necesario, por la especificación de la unidad.

Son admisibles los caracteres globales para especificar los ficheros; después de haber dado el comando se puede cambiar el directorio en uso, pero no se puede quitar el disco que contiene los ficheros a imprimir ni tampoco variar o borrar ninguno de los ficheros que están ya en cola de impresión. La impresora se mantiene ocupada hasta la terminación de la impresión y cualquier tentativa de utilizarla genera mensajes de error:

no preparado; error en dispositivo PRN:
Cancelar, Reintentar, Ignorar?

Cuando se utiliza por primera vez el comando, el MS-DOS visualiza la solicitud:

Nombre de la unidad de impresión [PRN]:

Si se pulsa la tecla de introducción, el nombre de la unidad asignada será PRN, pero se pueden dar también los nombres de otras unidades: LPT1, LPT2, LPT3, COM1, COM2, AUX, etc. Se elige así la unidad de salida deseada.

Si quisiéramos interrumpir la impresión, podemos volver a dar el comando PRINT/T que detiene la ejecución del comando borrando el fichero en impresión y los que están en cola.

El parámetro "/C" permite suprimir un fichero o una serie de fichero de la cola de impresión. Para hacerlo, tenemos que especificar "/C"; este parámetro borrará el fichero que le precede y todos los que le siguen en el mismo comando.

Por ejemplo:

A>PRINT B:FICHE1.LST/C FICHE2.LST A:FICHE3.LST

borrará los tres ficheros citados en el propio comando.

Por el contrario:

A>PRINT B:FICHE1.LST FICHE2.LST/C A:FICHE3.LST

añade el FICHE1.LST a la cola de impresión y suprime de la misma a los ficheros FICHE2.LST y FICHE3.LST.

- RECOVER - el "salvador"

Se trata de un comando que no se debe utilizar muy a menudo. Hay que tener en cuenta que aun cuando tratemos de respetar todas las normas de seguridad de los da-

tos y las relativas a la buena conservación de los discos, podemos tener un contratiempo en cualquier momento; este comando nos puede ser de gran ayuda.

Sirve para recuperar uno o la totalidad de los ficheros de un disco, o bien el directorio del disco. La recuperación de un fichero se produce, por supuesto, perdiendo los datos que estaban contenidos en los sectores deteriorados.

Este comando es de tipo *externo*, su formato es:

A>RECOVER B:DETERIORADO.BAT

o bien

A>RECOVER B:

El segundo formato sirve para recuperar todos los ficheros de un directorio y ha de utilizarse con cuidado porque el MS-DOS presupone que el directorio está irremediablemente deteriorado y, por consiguiente, trata de recuperar cualquier fichero que, a su juicio, pudiera ser todavía válido.

Para cualquier fichero recuperado, el MS-DOS crea un directorio que tiene la especificación del fichero con el formato

FILEnnnn.REC

en donde nnnn es un número progresivo.

- RENAME (REN) - un nuevo nombre

Tiene por objeto cambiar el nombre del fichero especificado por el nombre y la extensión que especifiquemos inmediatamente después.

Se trata de un comando de tipo interno y se le puede emplear con la forma abreviada REN. En las especificaciones de los ficheros podemos utilizar los caracteres globales. No obstante, sigue siendo válida la recomendación de probar primero con el comando DIR para la comprobación, evitando así desagradables e imprevistas consecuencias.

Veamos algunos ejemplos:

A>REN MODE.COM *.ESE
A>RENAME B:NOMBREA NOMBREB
A RENAME B:NOMBREA NOMBREB.TXT
A RENAME COMANDO?.TXT *.COM
A RENAME B:\PRIMERO\SEGUNDO\PRUEBA\PRUEBA2

- RESTORE - vuelve a poner en su lugar los ficheros

Es el comando que debe de utilizarse para restaurar los ficheros salvados con el comando BACKUP. Es de tipo externo y requiere como primer parámetro la unidad a partir de la cual queremos restaurar los ficheros, luego la unidad sobre la que queremos efectuar dicha restauración y, finalmente, la especificación del fichero.

También en este caso se pueden utilizar los caracteres globales:

Ejemplos:

A>RESTORE A: C:*.*/P
A>RESTORE A: C:*.COM
A>RESTORE A: C:*.*
A>RESTORE A: C:\PRIMERO\SEGUNDO\PRUEBA.XXX
A>RESTORE A: C:N/S

El último ejemplo contiene el parámetro "/S", que ordena la restauración de todos los subdirectorios salvados utilizando el mismo parámetro, pero con el comando BACKUP.

En el primer ejemplo se utiliza el parámetro "/P" para hacer que el DOS envíe un mensaje antes de restaurar los ficheros que están marcados como de sólo lectura (por ejemplo, IBMBIO.COM u otros ficheros clasificados como de aplicación). El mismo mensaje se envía antes de restaurar los ficheros que fueron actualizados después de haberse efectuado la copia de seguridad.

Podemos elegir así entre restaurar el fichero o renunciar a ello.

El comando RESTORE establece también un indicador que puede ser objeto de consulta con el comando IF en un fichero "Batch" (.BAT). Puede tomar los valores siguientes:

Ø restauración ejecutada de forma normal;
1 no se ha encontrado ningún fichero que restaurar;
3 interrumpida por comando del usuario;
4 interrumpida por error

- RMDIR (RD) - supresión de un directorio

Es necesario este comando para eliminar un subdirectorio del disco flexible o del disco duro.

Se trata de un comando de tipo interno y puede emplearse en la forma abreviada RD. Para poder suprimir un subdirectorio, éste debe estar completamente vacío. No se

puede borrar el directorio raíz ni el subdirectorio que está en uso.

A>DEL B:\PRIMERO\SEGUNDO *.*
(borramos todos los ficheros)
A>RD B:\PRIMERO\SEGUNDO
(destruimos el directorio)

• SYS - el sistema operativo

Se trata de un comando de tipo externo que transfiere el núcleo del sistema operativo desde el disco en donde reside al introducido en la unidad especificada.

Ejemplo:

A>SYS B:

transfiere el sistema operativo desde el disco A: al disco B:, que debe estar preparado para recibirlo.

Para preparar el disco es suficiente utilizar el comando FORMAT B:/S o bien FORMAT B:/B (vea el comando FORMAT).

Sirve también cuando se quiere transferir el sistema operativo a discos de programas de aplicación que se suministran sin el MS-DOS.

• TIME - la hora

Permite introducir o variar la hora establecida en el momento del encendido del ordenador. Para poderla cambiar es necesario pedir la hora. De este modo, todos los campos no especificados se ponen a cero.

Si no se especifica la hora, no se realizará ninguna variación. Se trata de un comando de tipo interno.

Si introducimos el comando

A>TIME

el DOS responde con el mensaje:

La hora actual es hh :mm :ss.cc
Introducir la nueva hora:

Es necesario respetar los delimitadores; entre las horas (hh), los minutos (mm) y los segundos (ss) existen dos puntos, mientras que un punto separa los segundos de las centésimas (cc).

• TREE - la estructura

Es un comando de tipo externo que sirve para obtener la visión completa de la organización existente en el disco de la unidad especificada. Si, además, especificamos el parámetro "/F" tendremos también la lista de todos los ficheros contenidos en los directorios.

A>TREE/F DIRMDA.LST

o bien:

A>TREE/F>PRN

(>PRN saca los resultados por la impresora) y en la impresora obtendremos:

LISTA CAMINO DIRECTORIOS
DEL VOLUMEN ???????????

Camino:	\PRIMERO
Subdirectorios:	SEGUNDO
Ficheros:	COMANDØ .PAX
	COMAND1 .PAX
	COMAND2 .PAX
Camino:	\PRIMERO\SEGUNDO
Subdirectorios:	ninguno
Ficheros:	COMANDØ .PAXT
	COMAND1 .PXT

Se observa a la salida la disposición en columna de la extensión, exactamente después de los ocho posibles caracteres del nombre del fichero.

• TYPE - el espía

Efectúa la visualización del contenido del fichero que se especifica. Es un comando de tipo interno y puede utilizarse solamente con los ficheros de tipo de texto, pues los caracteres existentes en los ficheros de tipo objeto (salida de compiladores) no son legibles en la pantalla.

Con este comando no se pueden utilizar los caracteres globales.

• VER - versión del S.O.

Sirve para comprobar la versión del sistema operativo que se está empleando. Es un comando de tipo interno.

- VERIFY - verificación de lo que se copia

  Vimos ya en el comando COPY cómo es posible hacer que el MS-DOS verifique si los datos que se acaban de escribir en el disco están escritos de modo legible. Con este comando, de tipo interno, dicha verificación se efectúa siempre y no solamente para el fichero que se está copiando.

  Con VERIFY ON se activa la verificación; si se quiere desactivar esta acción se puede escribir VERIFY OFF.

- VOL - nombre del disco

  Sirve para comprobar el nombre del disco introducido en la unidad especificada. Se trata de un comando de tipo interno.

CAPITULO VI

OTROS COMANDOS DEL MS-DOS

El mundo del MS-DOS es bastante rico, sobre todo después del gran salto cualitativo realizado con la versión 2.0.

En el capítulo anterior hemos pasado revista a los comandos más frecuentes; ahora vamos a completar nuestro lamentablemente rápido recorrido examinando otros comandos y echando un vistazo a los típicos comandos de utilidad.

Es preciso advertir que de estos últimos hay infinidad, desde el momento en que cualquiera puede desarrollarlos e introducirlos en el mercado abarcando necesidades especiales.

Comandos de configuración

Cuando se inicializa el MS-DOS, una de las primeras cosas que busca en el directorio-raíz es si existe el fichero CONFIG.SYS. Si lo encuentra, los comandos que contiene serán objeto de lectura y de interpretación; si se modifica después el fichero, la acción de los nuevos comandos tendrá efecto solamente en la próxima inicialización.

Damos a continuación la serie de comandos posibles:

- BREAK

  En condiciones normales, el MS-DOS comprueba si se han pulsado las teclas Ctrl-Break solamente durante las operaciones con las unidades periféricas y no cuando ejecuta funciones en memoria; esta es la situación de BREAK

OFF. Si queremos poder interrumpir la tarea del procesador cuando esté desarrollando cualquier función tendremos que escribir BREAK ON.

- BUFFERS

Cuando se inicializa el MS-DOS asigna en memoria dos zonas que utilizará para conservar los datos en tránsito antes de escribirlos en las unidades de disco o bien inmediatamente después de haberlos leído y antes de transmitirlos, uno tras otro, al programa de aplicación.

El número de buffers asignados por el sistema suele ser suficiente, pero en condiciones especiales podría ser de gran utilidad aumentar este número, sobre todo para optimizar los tiempos de ejecución.

En tal caso será suficiente escribir:

BUFFERS=nn

en donde "nn" es un número que puede llegar hasta un valor máximo de 99.

- DEVICE

Es posible crear programas para controlar unidades adicionales y grabarlos en disco. Cuando se quieren utilizar estas unidades con el fichero de control escrito por nosotros será necesario proporcionar al MS-DOS la indicación de qué fichero debe cargar para el control de las unidades; para hacer esto último será necesario especificar tantos comandos DEVICE= especificación de fichero como nuevos programas de control de unidades queramos utilizar:

DEVICE=PLOT.CON

- FILES

La versión 2.0 del MS-DOS ya no solicita a los programas de aplicación la creación de bloques de control para todos los ficheros que gestiona, puesto que los bloques (FCB - File control block=bloque de control de ficheros) se crean de forma automática por el sistema en el momento de la apertura del propio fichero. Ocupan una determinada zona de memoria que, en el momento del encendido, el MS-DOS establece en 8.

También en este caso podemos cambiar el número de ficheros abiertos y llegar hasta 99 con

FILES=nn

- SHELL

Es posible utilizar un procesador de los comandos que no sea COMMAND.COM especificando el nombre del fichero como parámetro del comando SHELL=especificación del fichero.

Los programas de utilidades

- LINK

El programa LINK sirve para agrupar varias piezas de un puzzle como, en este caso, es el programa final. En efecto se pueden crear programas, fragmento tras fragmento, y luego compilarlos o ensamblarlos y crear así módulos objeto en lenguaje máquina. Todos estos módulos deben agruparse luego para crear el programa final y esta es precisamente la tarea del programa LINK.

El MS-DOS suele tener como dotación solamente el intérprete BASIC avanzado, pero ni siquiera un compilador, por lo que será completamente inútil el programa LINK si no se adquiere un compilador o un ensamblador, tal como los de BASIC, COBOL, Pascal, FORTRAN, Macro Assembler, FORTH, C., etc. Son precisamente estos compiladores los que pueden crear las piezas de ese mosaico que el programa LINK, entre sus diversas funciones, logrará ensamblar.

A continuación damos un resumen de las diversas funciones desempeñadas por el programa LINK:

— Conexión de módulos objeto obtenidos por separado;
— búsqueda en los ficheros de biblioteca de las referencias externas que no están resueltas;
— resuelve las referencias cruzadas externas ("cross references");
— proporciona una lista de informaciones imprimibles en donde poder encontrar las referencias externas que se hicieron y los ocasionales mensajes de error;
— genera un módulo reubicable, que es posible situar en puntos diferentes de la memoria.

El programa utiliza y procesa una serie de ficheros:

Fichero	Extensión	Características
Objeto	.OBJ	salida de compilador
Biblioteca	.MAP	salida de compilador
De ejecución	.EXE	reubicable
VM	.TMP	provisional

para obtener al final un fichero en forma ejecutable con la extensión ".EXE".

La explicación de todas las funciones del programa LINK sería muy larga e inútil si no se tiene la intención de utilizar un compilador, y por ello remitimos a los lectores al manual del sistema operativo MS-DOS para todas las instrucciones nece cerias.

- DEBUG

También este programa está grabado entre los programas complementarios del MS-DOS en el segundo disco. Su utilización es muy particular; resulta recomendable para quienes tengan ya una cierta experiencia. Sirve para funciones tales como:

— crear un "ambiente" de prueba para poder efectuar comprobaciones acerca de la ejecución de los programas que se están tratando de diseñar. En efecto, es posible modificar el programa de forma directa y volverlo a ejecutar así de manera inmediata, sin pasar a través de una nueva fase de compilación;
— cargar en memoria cualquier fichero y modificarlo o bien limitarse a comprobarlo visualmente en la pantalla;
— poner en ejecución ficheros objeto que no se manipularon todavía por el programa LINK.

Veamos las funciones individuales con un poco más de detalle.

Para inicializar el programa es necesario escribir:

A>DEBUG B:COMAND1.LST parámetro1 parámetro2

Después del nombre del fichero se pueden especificar dos parámetros, que serán leídos por el fichero especificado. Por ejemplo, podemos cargar en memoria el comando DISKCOPY:

A>DEBUG DISKCOPY.COM A: B:

y los dos parámetros especificados son los que se utilizarían por el comando si lo quisiéramos ejecutar.

Los comandos que pueden emplearse en el programa DEBUG son todos ellos abreviados y de una sola letra (minúscula o mayúscula, no tiene ninguna importancia). Además, los diversos parámetros pueden separarse por delimitadores o bien no tener ninguna separación. El único delimitador obligatorio es el que está entre dos valores hexadecimales y son también válidos, por ejemplo, el espacio y la coma.

Para interrumpir la ejecución del comando proporcionado es necesario pulsar Ctrl-Break. A diferencia con lo que es habitual, estas teclas no interrumpen el programa DEBUG sino solamente el comando en ejecución en ese momento. Para interrumpir el programa DEBUG, y salir de este ambiente es necesario teclear Q (por Quit=abandonar).

Si introducimos un comando erróneo, el programa nos indica la posición en la cual se prevé que está el error. Cuando el programa DEBUG está preparado para recibir los datos visualizará su disponibilidad con un trazo (—).

Examinemos ahora, de forma sucinta, los diversos comandos.

— ASSEMBLE

Permite introducir programas escritos en lenguaje máquina utilizando la sintaxis del lenguaje estándar de los microprocesadores 8086 y 8088, con algunas peculiaridades normalizadas. Se puede definir también el punto a partir del cual queremos que se ensamblen las instrucciones.

— COMPARE

Permite la comparación de dos bloques de memoria entre sí; si existen diferencias, serán visualizadas especificando la dirección y el contenido.

— DUMP

Nos permite ver en la pantalla una parte de memoria en dos modos diferentes: a la izquierda en hexadecimal y a la derecha en los correspondientes caracteres ASCII. Si en la memoria hay caracteres que no se pueden visualizar, dichos caracteres se verán como un punto (.). Es posible especificar el punto de partida y también el de final.

— ENTER

Visualiza y/o cambia el contenido de los bytes. Es posible efectuar estas variaciones especificando la dirección de partida y, luego, la lista de los nuevos valores que queremos que sustituyan a los existentes en memoria.

Si no especificamos el valor de al menos un byte, el programa visualiza el byte correspondiente a la dirección especificada y espera a que se le dé el nuevo valor. Después de haber proporcionado el nuevo valor, tenemos que pulsar la barra espaciadora para ver el byte siguiente, y así sucesivamente. Cuando queramos grabar la dirección anterior tenemos que introducir un guión (—) y el programa visualizará el byte anterior. Se puede terminar con la simple pulsación de la tecla de introducción.

— FILL

Rellena las posiciones de memoria especificadas, proporcionando la dirección de partida y la de llegada y los valores correspondientes, uno tras otro. Si los valores indicados en la lista no son suficientes para rellenar todos los bytes incluidos entre los dos valores de partida y de llegada, el programa volverá a comenzar desde el principio de forma cíclica.

— GO

Pone en ejecución los programas que se han cargado, permitiendo su interrupción si hemos especificado la dirección en la que queremos que se interrumpa. También se pueden visualizar los registros, los indicadores y las instrucciones a ejecutar.

— HEXARITHMETIC

Ejecuta las operaciones de suma y resta de dos valores hexadecimales. La suma y la resta se visualizan a la vez.

— INPUT

Introduce y visualiza, en hexadecimal, un byte desde el punto de inserción especificado.

— LOAD

Permite la carga en memoria de un fichero o solamente de sectores de un disco; si se quiere, también es posible especificar la dirección de memoria de la que partir.

— MOVE

Transfiere el contenido de un bloque de memoria a otro sector de la misma. Se debe especificar la dirección de partida y de final del bloque que se quiere copiar y la dirección a partir de la cual se quiere iniciar la copia.

— NAME

Puede preestablecer los bloques de control de los ficheros que han de utilizarse con los comandos LOAD y WRITE; si el nombre del fichero se ha empleado cuando se activó el comando DEBUG no será necesario utilizar este subcomando.

— OUTPUT

Envía el byte a una dirección especificada para su inserción.

— QUIT

Interrumpe el empleo del programa DEBUG; por supuesto, si se quiere salvaguardar el programa que se está utilizando será necesario introducir primero el comando WRITE.

— REGISTER

Permite visualizar el contenido de los registros y cambiar sus contenidos. Además de los registros permite visualizar también los contenidos de los indicadores alfabéticos.

— SEARCH

Efectúa la búsqueda, en un bloque de memoria especificado con la dirección de partida y de final, de los caracteres especificados.

Si se encuentran los caracteres, se visualizará la dirección y el carácter contenido, y de no ser así, aparecerá solamente el guión trazo (—), que es el mensaje de solicitud del programa DEBUG.

— TRACE

Ejecuta una o varias instrucciones; si es más de una será posible especificar cuántas deben ejecutarse. La visualización puede interrumpirse en cualquier momento utilizando las teclas Ctrl-Numlock.

— UNASSEMBLE

Permite transformar el contenido de la memoria en una instrucción del tipo ensamblador. En la práctica, permite traducir el lenguaje máquina a instrucciones comprensibles del lenguaje ensamblador.

La visualización del programa se realiza tanto en caracteres hexadecimales como en caracteres ASCII.

— WRITE

Escribe en el disco el fichero que se ha manipulado en memoria y permite así guardar todas las modificaciones realizadas hasta ese momento.

• EDLIN

Es un programa que permite la creación y actualización de ficheros. En el ámbito técnico se le denomina "Editor de línea". Este término no nos complace en absoluto, pero no hemos conseguido encontrar la palabra adecuada, aunque, en teoría, podría ser "redactor", puesto que realiza una labor parecida a la de quien escribe, revisa y corrige los artículos. En todo caso "editor" es un término ya extendido y aceptado.

Indicamos a continuación las funciones desempeñadas por EDLIN:

— permite crear nuevos ficheros en formato de texto o fuente, es decir, los programas todavía no ensamblados ni compilados y, por supuesto, permite también su grabación;
— actualiza los ficheros ya existentes y permite salvar tanto el fichero actualizado como el fichero original, cambiando la extensión del original por .BAK;
— permite borrar, insertar y visualizar cualquier línea que se quiera;
— busca un texto en todo el fichero; lo puede suprimir o sustituir y lo hace en una o en varias líneas.

Todos los ficheros creados con el editor de línea tienen una longitud de registro no superior a los 253 caracteres. Se trata de registros variables (no todos tienen la misma longitud) lo que permite ahorrar mucho espacio en la escritura de ficheros que tienen los registros de longitud muy distinta entre sí.

Cuando se utiliza EDLIN se verán en la pantalla las líneas de los ficheros precedidas por un número, progresivo, que identifica durante el proceso a cada una de las líneas. Este número no se salva cuando el fichero es objeto de escritura. Se actualiza durante el proceso de modo que si se efectúan adiciones o supresiones de líneas, todas estarán siempre numeradas de forma consecutiva, sin saltos.

EDLIN tiene como carácter de solicitud de introducción de los datos el asterisco (*). Otra característica es que nunca carga la memoria en más del 75 por 100, de modo que se tenga siempre un cierto margen de maniobra. No obstante, se pueden actualizar ficheros mucho más grandes efectuando la actualización en la parte ya cargada, grabar luego estas primeras líneas en el disco, cargar otro trozo y así sucesivamente hasta el final.

Se puede iniciar la creación de un fichero con un comando del tipo:

A>EDLIN B:FICHEROP.TXT

A continuación se verá aparecer el mensaje:

Nuevo fichero
*_

En este punto, el programa pide un comando que debe ser, en esta fase inicial, obligadamente del tipo I, es decir de inserción de nuevas líneas.

A partir de este momento será posible introducir todas las líneas que queramos, una tras otra. Para acabar una línea hay que pulsar la tecla de introducción.

Una vez terminada la edición será posible grabar el fichero con el empleo del comando E.

Por el contrario, si el fichero existiera ya, el programa cargará el fichero en la memoria (interna) y proporcionará el mensaje siguiente:

*_

Si el fichero no se ha cargado en su totalidad por falta de memoria RAM no aparecerá el mensaje y se pondrá de manifiesto por la aparición del asterisco sólo. Los comandos del programa se indican a continuación:

— A - Adición de líneas; se especifica el número de las líneas a añadir seguido por la letra A

por ejemplo: 5A.

— C - Copia líneas; se especifica el número de la línea que se quiere copiar o el número de las líneas que se quieren copiar en el formato "desde/a"; luego, separado también por una coma, el número de la línea en la cual se copiará o a partir de la cual se copiarán las líneas y, finalmente, la letra C

por ejemplo: 3, 5, 9C.

— D - Borrado de líneas; se especifican las líneas a borrar en la forma "desde/a" seguidas por la letra D.

por ejemplo 3,5D.

— N - Edición de líneas; permite llamar la línea a corregir o a comprobar, visualizándola. Para hacerlo es suficiente escribir el número de la línea que nos interesa.
— E - Final de edición; se comunica así al programa que hemos terminado la tarea y que puede escribir el fichero en disco
— I - Inserción de línea; introduce una línea inmediatamente antes de la línea especificada

por ejemplo: 3I.

— L - Lista las líneas; permite visualizar en la pantalla las líneas que nos interesan, una o más de una, proporcionando el número de línea o el intervalo "desde... a"

por ejemplo: 5, 25L.

— M - Desplazamiento de líneas; sitúa las líneas especificadas por un intervalo (desde... a...) delante de la línea especificada como tercer parámetro

por ejemplo: 2,5,108M.

— P - Paginación; permite recorrer el fichero en páginas de 23 líneas cada una, o menos, según lo que se indique. El primer parámetro indica a partir de qué línea, y el segundo de cuántas líneas debe efectuarse la paginación; si falta se supone que son 23

por ejemplo: 3,15P.

— Q - Final de proceso; concluye el trabajo como si hubiéramos estado "sólo de broma" y hubiéramos querido solamente ver el fichero. Antes de "tirar todo" nos hará una pregunta:

*Q
Quiere terminar el trabajo (S/N)?—

Por supuesto, podemos responder N y continuar nuestro trabajo.
— R - Sustitución de texto; según se indica literalmente en el manual, sustituye el contenido de la primera cadena en el intervalo de líneas especificadas por el contenido de la segunda cadena. Como mejor explicación hemos recurrido a un ejemplo:

1,228 R mario

En este caso, queremos buscar desde la línea 1 a la línea 228 la palabra "mario" y deseamos sustituirla por la palabra "Mario". Inmediatamente después de haber escrito lo anterior pulsamos la tecla de función F6 y escribimos:

Mario

Después de haber pulsado la tecla de introducción se obtiene la visualización de todas las líneas que han sufrido la variación de la palabra "mario" por la de "Mario".
— S - Búsqueda de texto; se puede buscar un texto a partir de las líneas especificadas

por ejemplo 1,228 S mario.

— T - Transferencia de líneas: transfiere el contenido del fichero que especifiquemos después del carácter T al fichero que se está procesando a partir de la línea especificada como primer parámetro

por ejemplo: 3 T B:FILE1.TXT transfiere
el fichero FILE1.TXT

grabado en el disco B: a la memoria a partir de la línea 3.
— W - Escritura de líneas; transfiere al disco el número de líneas especificado, a partir de la primera (por ejemplo: 228 W).

CAPITULO VII

LOS FICHEROS "BATCH"

Se dice que un fichero es de tipo "batch" (o por lotes) cuando contiene una secuencia de comandos que se leen en un determinado orden y se ejecutan, uno a uno, por el MS-DOS. Si el fichero "batch" llamado no se encuentra en el directorio en uso, el MS-DOS lo busca en los otros directorios que se han especificado con el comando PATH.

La extensión de los ficheros por lotes ("batch") debe ser siempre ".BAT" y para ejecutarlos basta introducir el nombre sólo, sin la extensión. Para interrumpir la ejecución es necesario pulsar las teclas Ctrl y Break; se visualizará así la pregunta:

Quiere terminar el trabajo (S/N)?

Si respondemos S el trabajo se interrumpirá de forma definitiva; si pulsamos N se interrumpirá solamente el comando en ejecución y el trabajo se reanudará con el siguiente.

Si durante la ejecución se extrae el disco que contiene los ficheros "batch" de la unidad que lo contenía, el sistema solicita su nueva inserción para poder leer otro comando. Se acepta como comando el nombre de otro fichero "batch", de tal modo que se puedan ejecutar una serie de cadenas de comandos. Hay que tener cuidado, sin embargo, con no crear "nidificaciones".

Otra característica importante es que el sistema recuerda en qué directorio se ha iniciado el fichero "batch", por lo que es posible cambiar el directorio en el interior del fichero de los comandos como y cuando se quiera.

AUTOEXEC.BAT

El fichero AUTOEXEC.BAT es un fichero especial que tiene una función bien específica. Cuando el procesador de los comandos toma el control después de la inicialización del MS-DOS, lo primero que hace es comprobar si en el directorio existe este fichero y, si lo encuentra, lo ejecuta. Por ello se pueden ejecutar comandos diferentes para cada disco que tenga el MS-DOS residente. Un ejemplo podría ser:

```
ECHO OFF
KEYBSP
REM*** Disco MS-DOS para composición de textos***
DATE
TIME
PROMPT $P$ $T—$—Que hago?—$—
```

En la práctica, el fichero del ejemplo no hace otra cosa que establecer el teclado español, la fecha, la hora y el mensaje de solicitud del sistema. Este conjunto de comandos se puede comprender mejor después de haber leído la descripción de los comandos avanzados del MS-DOS (capítulo 2).

Otro ejemplo clásico es el "autolanzamiento" de un programa de aplicación contenido en el disco en el momento del encendido. En el supuesto de que se quiera inicializar automáticamente un determinado programa de tratamiento de textos y admitido que se le denomina EDIT, podemos establecer estas condiciones como sigue:

```
A>COPY CON: AUTOEXEC.BAT
```

Este comando permite introducir líneas de comando directamente desde el teclado en el fichero AUTOEXEC.BAT.

Inmediatamente después, introduciremos el nombre del programa de aplicación

```
EDIT
```

Pulsamos la tecla de introducción, luego la tecla de función F6 y, finalmente, otra vez la tecla de introducción. De este modo hemos creado otra línea en el fichero y todas las veces que insertemos el disco y encendamos el sistema, el MS-DOS cargará automáticamente el programa EDIT.

Es posible proporcionar al fichero "batch" unos parámetros externos que puedan hacer que el mismo fichero se comporte de manera diferente.

Un parámetro se individualiza, dentro de un fichero "batch", con el símbolo % seguido por un número (por ejemplo: %1, %2, etc.).

A un número igual corresponde, por supuesto, el mismo elemento, que puede aparecer varias veces en el fichero "batch". Veamos un ejemplo:

```
REM Cuando se solicite el disco A: inserte
REM el DISCO RESERVA A—B.B.I.
COPY CAPIT%1.TXT B:*.*/V
REM A la solicitud de insertar el disco B: introduzca
REMel DISCO RESERVA B—B.B.I.
COPY CAPIT%1.TXT B:CAPIT2%1.TXT
```

De este modo, cuando hayamos acabado una parte del trabajo estaremos precavidos contra los borrados o pérdidas accidentales, indicando el capítulo que acabamos de actualizar o de escribir en dos discos de reserva con la simple especificación del nombe del fichero ".BAT" que, en este caso, hemos llamado de forma muy prosaica: SALVA.BAT. En la práctica, cada vez que queramos ejecutar la doble copia teclearemos:

```
A>SALVA n
```

en donde "n" es el número del capítulo en cuestión.

Dicho dato se sustituye en todas las utilizaciones del parámetro %1 dentro de SALVA.BAT.

Es evidente que los ficheros deben llamarse todos ellos de la forma CAPITn.TXT.

Se pueden especificar hasta 10 parámetros dentro de un fichero ".BAT"; si queremos especificar más será necesario emplear el subcomando SHIFT.

Subcomandos admisibles en un fichero tipo Batch

- ECHO - El subcomando en cuestión activa o desactiva la visualización de los comandos MS-DOS que se ejecutan en un fichero batch. El formato del comando es muy simple, puesto que basta con proporcionar el parámetro ON para la visualización o el parámetro OFF para la no visualización.

- FOR - Permite le ejecución iterativa de los comandos. No está permitida la nidificación (un FOR dentro de otro).

  Un pequeño ejemplo puede servir de aclaración:

```
FOR %%a IN (prueba1 prueba2 prueba3) DO print %%a
```

el comando establece la variable "a" en secuencia con el contenido del grupo encerrado entre paréntesis y luego ejecuta el comando que sigue al DO. En la práctica el comando envía a la cola de impresión los tres ficheros contenidos entre paréntesis.

- GOTO - Transfiere el control a la "etiqueta" especificada, con lo que permite saltar la ejecución de algunos comandos. La etiqueta se especifica inmediatamente después del subcomando y debe definirse en el fichero "batch" precedida por dos puntos (:). La etiqueta puede ser una larga serie de caracteres, pero solamente los ocho primeros serán significativos. Si una etiqueta no es llamada por ningún subcomando GOTO no habrá ningún control; esto permite utilizar las etiquetas como líneas de comentario que no se visualizan en el momento de la ejecución de los comandos.

- IF - Permite la ejecución condicionada de los comandos. Veamos con algunos ejemplos qué parámetros pueden utilizarse para verificar algunas condiciones:

  — IF EXIST CAPIT1.BAK DEL CAPIT1.BAK
  En este caso, si existe la copia de reserva del fichero CAPIT1.TXT se borrará, y si no existe, no se ejecutará el comando DEL
  — IF NOT EXIST CAPIT1.BAK GOTO COPIA
  ....
  :copia
  COPY CAPIT1.TXT CAPIT1.BAK
  En este caso, si la copia NO existe se transmitirá el control al comando que está inmediatamente después de la etiqueta: copia.
  — IF%1 = = SECRETO GOTO ADELANTE
  ECHO Palabra clave no adecuada termina la ejecución
  GOTO final
  El parámetro suministrado se compara con la palabra "secreto"; si son iguales proseguirá la ejecución pasando el control a la etiqueta "adelante", si no fueran iguales, se interrumpirá el proceso.
  — IF ERRORLEVEL 1 ECHO final ejecución por error
  Este comando debe ir seguido siempre por otro, que suele ser el nombre del programa que establece el código de error de modo significativo. En este caso, el programa debe poner a 1 el indicador de error si algo fue mal durante la ejecución, de modo que al verificarse la condición se ponga de manifiesto el mensaje. Por supuesto, los demás valores no hacen terminar la ejecución del fichero "batch".

- SHIFT - Permite proporcionar al fichero "batch" más de 10 parámetros; para conseguirlo será necesario suministrar todos los parámetros en el comando del fichero "batch" y luego utilizar el subcomando para hacer que los parámetros se desplacen a la izquierda una posición por cada subcomando SHIFT proporcionado.

  Para sustituir los 10 primeros parámetros por otros 10 tendremos que escribir 20 parámetros y luego, una vez utilizados los 10 primeros, o cuando sea preciso, dar 10 veces el subcomando SHIFT.

- PAUSE - Suspende el proceso hasta que se pulsa una tecla cualquiera. Es posible escribir un comentario

  PAUSE Introducir disco A y pulsar cualquier tecla

- REM - Permite visualizar los mensajes escritos en un fichero "batch" durante la ejecución del mismo.

Ejemplos de ficheros .BAT

- Para inicializar un disco:

```
FORMAT B:/B
REM insertar en la unidad A: la versión del MS-DOS
REM que se quiera usar
PAUSE pulsar ENTER cuando esté preparado
SYS B:
COPY AUTOEXEC.BAT
COPY COMMAND.COM B:
COPY MODE.COM B:
COPY WTDATIM.COM B:
COPY KEYBSP.COM B:
Pause — inicialización terminada
```

- Ejemplo de aplicación

```
ECHO OFF
ASSIGN B=A
MODE 40
IF EXIST INDEX.TMP GOTO OK
GOTO NEXIST
```

```
:OK
ERASE INDEX.TMP
:NEXIST
PRELAB
MODE 80
ELAB TEXTO.CMD
IF EXIST INDEX.TMP GOTO OKX
GOTO NEXISTX
:OKX
ELABIND
:NEXISTX
:LABEL solamente para comentario: restauración de las
asignaciones
ASSIGN
```

- Introducción en el disco fijo de un producto:

```
REM
CLS
C:
REM INTRODUCCION PRODUCTO EN IBM XT
REM
REM ESPERAR
CD
MD C:   PRODUCTO
CD   PRODUCTO
COPY A:BASRUN.EXE
COPY A:FIL*.*
COPY A:FAL*.EXE
COPY A:STIPO.EXE
COPY A:TEST
COPY A:EJEMPLOS.*
CLS
:DISCOB
REM
REM INSERTAR EL SEGUNDO DISCO EN LA UNIDAD A
PAUSE
IF EXIST A:DISCOB GOTO BDISCO
REM************************
REM            ATENCION
REM
REM EL DISCO INTRODUCIDO NO ES EL DISCO B
REM
REM CORREGIR
REM************************
GOTO DISCOB
```



```
:BDISCO
COPY A:FILEB*.*
CD
CLS
```

- Para el final hemos dejado lo mejor, un fichero "batch" que permite copiar una serie de programas sin olvidarse de ninguno.

```
CHKDSK
PAUSE INTRODUCIR EL DISCO ORIGEN EN LA
UNIDAD A:
CHKDSK B:
REM*SIRVEN UNOS 70 K BYTES LIBRES PARA
REM*COPIAR TODOS LOS FICHEROS DE ESTE
PRODUCTO
PAUSE INTRODUCIR EL DISCO EN EL QUE SE QUIERE
COPIAR
EN LA UNIDAD B:
REM*COMIENZA LA COPIA DE SEGURIDAD DEL
PRODUCTO
REM***COPIA DE LOS PROGRAMAS***
COPY A:MENU.BAS B:
COPY A:PINCO.BAS B:
COPY A:INFORMES.BAS B:
COPY A:SAVE.BAS B:
COPY A:CREA.BAS B:
COPY A:DOC.BAS B:
REM***COPIA DE LOS FICHEROS DE HELP***
COPY A:MHELP.TXT B:
COPY A:AHELP.TXT B:
REM***COPIA DE LOS FICHEROS BAT***
COPY A:MIOCOPY.BAT B:
COPY A:AAAS.BAT B:
REM***COPIA DOCUMENTACION***
COPY A:SYSDOC.TXT B:
REM***COPIA DE LOS FICHEROS DE ERROR***
COPY A:ERRMSGS B:
REM*COPIA EJECUTADA DE FORMA CORRECTA
```

CAPITULO VIII

COMANDOS AVANZADOS DEL MS-DOS

Redireccionamiento y transferencia de datos

Veamos ahora, basándonos en una exposición enriquecida con pequeños ejemplos prácticos, las funciones de redireccionamiento y "piping" (transferencia).

Como se dijo ya en el primer capítulo, las de redireccionamiento, representadas por los signos "<" (que expresa redireccionamiento en la entrada) y ">" (que expresa redireccionamiento en la salida), sirven para cambiar la procedencia y el destino, respectivamente, de la entrada y de la salida con respecto a la normalmente supuesta.

En condiciones normales, los datos de entrada para un programa de gestión de datos se extraen del teclado. Dicha asignación es automática, pero, a veces, sería de más utilidad y menos costoso utilizar siempre un fichero en el que se variaran solamente algunos datos.

En la versión 2.0 del MS-DOS podemos asignar, como unidad de entrada, un fichero en lugar del teclado mediante:

```
A>PROGMIO<B:DATINPUT.TXT
```

En este fichero tenemos que escribir con toda exactitud todo lo que hubiéramos tenido que escribir en el teclado. En efecto, si el programa no encuentra ningún parámetro o dato que le sirva al sistema se parará, porque el MS-DOS ya no estará en condiciones de resolver el problema; por ello debemos comprobar el fichero antes de su utilización.

La otra posibilidad es la de cambiar la salida estándar, que

suele ser la pantalla, asignando esta función a la impresora o bien a un fichero

A>DIR>PRN

Vimos anteriormente los nombres reservados por el MS-DOS para los dispositivos en la descripción detallada del comando COPY y hemos visto también que PRN es el nombre de la impresora. Con el comando anterior hemos solicitado al sistema que envíe la lista del directorio desde la pantalla a la impresora.

A>DIR>RAIZ.DIR

En este caso, hemos pedido la creación de un fichero que contenga la lista del directorio-raíz.
De este modo podemos asignar de forma diferente las diversas funciones.
Con la función de transferencia de datos ("piping") el MS-DOS nos permite también transferir los mismos datos que se hubieran enviado a la pantalla a otro programa como si procedieran del teclado.
Si tenemos, por ejemplo, un programa que nos proporciona algunos datos que normalmente tomamos e introducimos por el teclado cuando "lanzamos" un segundo programa, resulta inútil dicho tecleado intermedio de esos datos; simplemente se puede escribir:

A>PROG1|PROG2

La transferencia de datos de un programa a otro la realiza el MS-DOS escribiendo ficheros temporales con el nombre:

%PIPEx.$$$

Es necesario comprobar que dichos ficheros no se borran durante la ejecución de nuestros programas. La transferencia de datos consiste en el redireccionamiento automático de las unidades de entrada y de salida y se ejecuta mediante la barra vertical situada entre los dos nombres de programa.
El siguiente ejemplo, aunque introduce el comando SORT, que no hemos considerado todavía, puede servir para aclarar mejor el empleo del "piping".

A>DIR|SORT>RAIZ.DIR

El comando DIR pasa al comando SORT la lista de los ficheros del directorio, SORT los pone en orden y los direcciona hacia el fichero RAIZ.DIR (Fig. 4 del capítulo 1).

El manual del MS-DOS llama "comando-filtro" a un comando que, una vez recibidos datos a la entrada, los modifica y los restituye, de algún modo diferente, a la unidad de salida estándar (o modificada por redireccionamiento). Los comandos-filtro existentes en el MS-DOS son tres, pero podemos escribir tantos como queramos simplemente considerando que tenemos que recibir datos a la entrada desde la unidad de entrada y escribirlos en la unidad de salida.
Los tres comandos se describen en este mismo capítulo; son los denominados SORT, FIND y MORE.
Los veremos, como es habitual, en la lista alfabética de los comandos especiales.

CTTY

Permite cambiar las unidades estándar de entrada y de salida asignando otras o bien restaurando las consideradas estándar. Se trata de un comando de tipo interno. Admite como parámetros solamente nombres reservados. Ejemplos:

A>CTTY AUX
A>CTTY CON

El primer comando asigna unidades diferentes a las estándar, mientras el segundo restaura todo el conjunto.

EXE2BIN

Los ficheros con extensión ".COM" y ".EXE" actúan como ficheros-comando, con la salvedad de que los segundos no tienen un "trazado" óptimo; el programa EXE2BIN tiene precisamente el cometido de transformar los ficheros de tipo ".EXE", que no están optimizados en ficheros de formato ".COM", con lo que se ahorra espacio en la escritura de dichos ficheros en un disco y también se ahorra tiempo cuando se cargan en memoria.
Se trata de un comando de tipo externo y tiene la posibilidad de especificar dos parámetros: el primero identifica el fichero que debe transformarse y el segundo identifica el nombre del fichero del tipo ".COM" que debe crearse.
Si no se especifica la extensión del primer parámetro, el programa supone por defecto la extensión ".EXE", mientras que si no se especifica la extensión del segundo parámetro el fichero se escribirá con la extensión ".BIN".

Si no especificamos luego el nombre del fichero de salida, se supondrá que es igual al nombre del fichero de entrada.

Para obtener programas reubicables (.COM) sin problemas, se requieren, además de las nociones de programación, que se conozca el programa que efectúa las conexiones (LINK) y también alguna información sobre los registros, la distribución del espacio de memoria, etc.

EL programa EXE2BIN reside en el disco complementario del MS-DOS.

FIND

Este comando-filtro envía a la unidad de salida estándar todas las líneas del fichero, especificado como último parámetro, en las cuales el comando encuentra correspondencia entre los caracteres en ellas contenidos y los especificados en la cadena proporcionada como parámetro y encerrada entre comillas.

Se trata de un comando de tipo externo que no admite el empleo de caracteres globales en la especificación del fichero. Para controlar este comando hay tres parámetros:

- /V - Con este parámetro solicitamos al comando FIND que visualice todas las líneas que no contengan los caracteres especificados en la cadena.
- /C - Este parámetro nos permite saber cuántas veces son objeto de ocurrencia los caracteres de la cadena en el fichero, sin visualizar las líneas que los contienen.
- /N - Permite visualizar en la pantalla el número relativo de las líneas que contienen la cadena especificada. El número se pone entre corchetes delante de la propia línea.

Es posible realizar la búsqueda también en varios ficheros simplemente continuando la escritura de otros nombres después de la especificación del primer fichero.

Ejemplo:

A>DIR A:|FIND ".TXT"

Sirve para obtener la lista de todos los ficheros que, en el directorio A:, contienen la cadena ".TXT".

MORE

Es un comando-filtro que transfiere a la unidad estándar de salida toda una pantalla llena de datos, leyéndolos desde el teclado o por la unidad estándar activa en ese momento.

Se trata de un comando de tipo externo que, después de haber llenado toda la pantalla, envía el mensaje:

--Más--

Después de haber leído lo que nos interesa, podemos pulsar cualquier tecla para que se visualice una nueva pantalla llena de datos. En la práctica, se nos permite leer los ficheros página a página.

PROMPT

Es el comando que nos permite "personalizar" el mensaje de solicitud del sistema. Es de tipo externo. El mensaje que se escribe después del comando se supone que es el nuevo mensaje de solicitud del sistema.

Si no se especifica ningún parámetro después del mensaje, se utilizará el mensaje habitual (como "A>").

Se pueden introducir en el texto algunos mensajes especiales en la forma $n, en donde "n" puede tomar los valores siguientes:

- t - si se especifica visualizará la hora;
- d - en este caso visualiza la fecha;
- p - presenta el directorio en uso de la unidad supuesta por defecto;
- v - especifica el número de la versión;
- n - proporciona la unidad de disco supuesta por defecto;
- g - visualiza el carácter ">";
- l - para el carácter "<";
- b - para el carácter "|";
- q - para el carácter "=";
- h - efectúa un retroceso borrando el carácter anterior;
- e - carácter ESCape (desconexión);
- — = llega al final

Cualquier otro carácter no tendrá ningún efecto y se considerará como carácter nulo. Veamos ahora algunos ejemplos:

A>PROMPT qué hora es?
A>PROMT qué hago?

A>PROMT disco supuesto $p—
A>PROMT Hora = t—Fecha $d
A>PROMPT $X; introducción del comando

El primero y el segundo establecen solamente el mensaje; el tercero, además de establecer el mensaje, escribe la unidad supuesta por defecto, y el cuarto escribe en la primera línea la hora y en la segunda la fecha, ambas precedidas por el mensaje correspondiente.

En el quinto caso queremos utilizar como carácter del mensaje el delimitador de punto y coma (;) como primer carácter de mensaje y no como carácter de separación; por ello hemos puesto delante el delimitador del mensaje $X, que genera un carácter nulo, de modo que el delimitador sea considerado como el primer carácter del mensaje y no como separador entre dos parámetros.

Si queremos restaurar todo porque, a la postre, el mensaje estándar del sistema es el que más nos satisface, siempre podremos escribir el comando:

A>PROMPT ng

SET

Es un comando que puede utilizarse para almacenar en el ámbito operativo, en un bloque de memoria, nombres y cadenas de parámetros asociados a los propios nombres.

Se trata de un comando de tipo interno que sirve para que los programas de aplicación encuentren algunos parámetros establecidos con anterioridad, de modo que, por ejemplo, puedan saber en qué unidad han de ir a buscar los ficheros que deben utilizar.

Si proporcionamos un comando SET seguido por un nombre sólo, significa que queremos extraer de la memoria el propio nombre y los parámetros asociados.

Por el contrario, si tecleamos solamente SET queremos que el MS-DOS nos indique cuáles son los nombres que ha almacenado.

También los comandos PROMPT y PATH sitúan parámetros en la memoria del ambiente operativo, con la diferencia de que no tienen necesidad del comando SET, puesto que lo hacen todo por sí mismos.

Otro nombre que encontramos si tecleamos solamente SET es el que contiene el punto de conexión para poder recargar el fichero COMMAND.COM por parte del sistema si, por cualquier motivo, el procesador de los comandos se hubiera eliminado; será, por ejemplo: COMSPCE=A:NCOMMAND.COM.

Con este comando el MS-DOS expande una determinada zona de memoria para contener los nombres; no podrá hacerlo si hemos cargado ya como residente en la memoria un programa de aplicación.

SORT

Es el último comando-filtro del MS-DOS; hemos hablado de este comando al comienzo del capítulo y sabemos ya que transfiere los datos que lee desde la unidad estándar de entrada a la unidad estándar de salida después de haberlos ordenado.

Se trata de un comando de tipo externo que procede a ordenar los ficheros que están especificados o, más bien, el fichero que está especificado, en orden ascendente utilizando la secuencia de comparación de los caracteres ASCII.

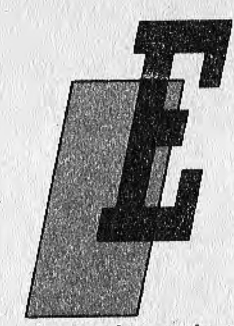
Es posible obtener el orden decreciente especificando el parámetro "/R", mientras que si no queremos iniciar la selección en la comuna 1 podemos especificar cuántas columnas han de saltarse y especificar "/+x"; precisamente "x" es el número de la columna a partir de la cual se debe comenzar a ordenar. Se pueden seleccionar ficheros con una longitud no superior a 63k.

Ejemplo:

A>SORT/R/+12<ASCENDE.NTE>DESCENDE.NTE

Queremos ordenar de modo descendente el fichero ASCENDE.NTE y el resultado escribirlo en el fichero DESCENDE.NTE. La ordenación debe comenzar en la columna 12.

APENDICE



En las tablas que damos a continuación las letras encerradas entre paréntesis pueden omitirse [por ejemplo, REN(AME) puede escribirse tanto de forma completa como en la forma abreviada REN].

En la columna marcada con I/E las dos letras identifican el tipo de comando: la letra I indica que el comando es de tipo interno (residente), mientras que la letra E significa que el comando es de tipo externo (transitorio, programa de utilidad).

Las informaciones encerradas entre corchetes no tienen por qué suministrarse. Si se quieren introducir no deben emplearse los corchetes.

Cuando las palabras están separadas por la barra vertical (|) significa que es posible elegir entre las dos palabras que aparecen a la derecha y a la izquierda de dicha barra.

Los signos de puntuación (. = ? : / N) deben respetarse, es decir, no es posible omitir ninguno de los signos indicados (solamente la barra vertical "|" y los corchetes no deben indicarse).

Cuando se especifica el parámetro "d:" significa que es necesario aclarar la unidad de disco que se quiere emplear (si se omite, el MS-DOS, por defecto de información, utilizará la unidad supuesta).

La palabra "camino" debe ser sustituida, si fuera necesario, por la especificación de los nombres de los directorios separados por barras invertidas (también el nombre de fichero final debe estar separado por la barra invertida).

Tabla compendio de los comandos del MS-DOS

Comando	I/E	Objeto	Formato
ASSIGN	E	Direcciona la búsqueda en unidades diferentes	ASSIGN [x=y[...]]
BACKUP	E	Crea una copia de reserva de los ficheros en disco duro.	BACKUP [d:][camino] [nombre fichero] [ext] d: [/S][/A][D:mm-dd-aa]
BREAK	I	Determina cuándo debemos antender y cuándo no, la interrupción generada por la pulsación de CTRL-BREAK	BREAK[ON/OFF]
CHDIR	I	Cambia el directorio utilizado en ese momento	CHDIR[[d:]camino] o bien CD[[d:]camino]
CHKDSK	E	Comprueba el disco y visualiza los resultados	CHKDSK[d:][nombre del fichero] [.ext][/F][/V]
CLS	I	Borra la pantalla	CLS
COMP	E	Efectúa una comparación entre ficheros y visualiza el resultado	COMP [d:][camino] [nombre fichero.ext][d:] [camino][nombre fichero[.ext]]
COPY	I	Copia los ficheros	COPY [/A][/B] [d:][camino]nombre fichero[.ext][/A][/B] [d:][camino][nombre fichero[.ext]][/A][/B][/V] o bien COPY [/A][/B] [d:][camino]nombre fichero[.ext][/A][/B] [+][d:][camino] nombre fichero [.ext] [/A][/B]...] [d:][camino][nombre fichero[.ext]][/A][/B][/V]
DATE	I	Establece la fecha	DATE [mm-dd-aa]
DEL	I	Ver ERASE	
DIR	I	Lista los nombres de los ficheros de la unidad solicitada	DIR[d:][camino][nombre, fichero[.ext]][/P][/W]
DISKCOMP	E	Compara los discos	DISKCOMP[d:][d:][1][/B]
DISKCOPY	E	Copia los discos	DISKCOPY[d:][d:][/1]

Comando	I/E	Objeto	Formato
ERASE	I	Borra los ficheros eliminando sus referencias en el directorio	ERASE[d:][camino] [nombre del fichero[.ext]] Puede utilizarse como DEL
FORMAT	E	Formatea los minidiscos	FORMAT[d:][/S][/1] [/8][/V][/B]
GRAFTABL	E	Carga la tabla de los caracteres gráficos	GRAFTABL
GRAPFICS	E	Imprime la pantalla con caracteres gráficos	GRAPHICS
KEYBnn	E	Carga el programa de soporte para los teclados no americanos	KEYBUK KEYBGR KEYBFR KEYBIT KEYBSP
MKDIR	I	Crea un directorio de nivel inferior (subdirectorio)	MKDIR[d:]camino o bien MD[[d:]camino]
MODE	E	Establece el modo de funcionamiento para la impresora y la pantalla	MODE LPT=:[n][,[m][,p]] o bien MODE n o bien MODE[n],m[,T] o bien MODE COMn:baud[,paridad[[,bits__dato[,bits__parada[,P]]]] o bien MODE LPT=:=COMn
PATH	I	Camino de búsqueda de los comandos y los ficheros "batch" en los directorios	PATH[[d:]camino[[;d:]camino]...]
PRINT	E	Pone en cola e imprime los ficheros especificados	PRINT[[d:][nombre del fichero [.ext]][/T][/C][/P]...]
RECOVER	E	Recupera ficheros de disquetes o del disco rígido	RECOVER[d:][camino] nombre del fichero[.ext] o bien RECOVER D:
RENAME	I	Cambia el nombre a los ficheros	REN(AME)[d:][camino] nombre del fichero[.ext] nombre del fichero[.ext]
RESTORE	E	Efectúa la restauración de los ficheros desde disquetes a disco duro.	RESTORE d:[d:][camino] nombre del fichero[.ext] [/S][/P]

Comando	I/E	Objeto	Formato
RMDIR	I	Borra un directorio de nivel inferior (subdirectorio)	RMDIR[d:]camino o bien RD[d:]camino
SYS	E	Transfiere el MS-DOS	SYS d:
TIME	I	Establece la hora	TIME[hh:mm:ss.xx]
TREE	E	Visualiza todos los caminos existentes (estructura en árbol)	TREE[d:][/F]
TYPE	I	Visualiza el contenido de un fichero	TYPE[d:][camino] nombre del fichero[.ext]
VER	I	Visualiza la versión del sistema.	VER
VERIFY	I	Verifica, en escritura, los datos en el disco	VERIFY[ON/OFF]
VOL	I	Visualiza el nombre asignado al disco	VOL[d:]

Comandos MS-DOS para proceso por lotes (BATCH)

Comando	I/E	Objeto	Formato
(Batch)	I	Ejecuta los ficheros por lotes "Batch"	[d:] nombre del fichero [parámetros]
ECHO	I	Impide/permite la visualización de los comandos que aparecen	ECHO[ON|OFF][mensaje]
FOR	I	Ejecución iterativa de comandos	FOR % % variable IN (grupo)DO comando
GOTO	I	Transfiere el control a la línea indicada por la etiqueta	GOTO etiqueta
IF	I	Ejecución condicional de programas-comandos	IF NOT comando condicional
PAUSE	I	Produce una pausa en el sistema	PAUSE[comentario]
REM	I	Visualiza comentarios	REM[comentario]
SHIFT	I	Desplaza las líneas de comandos	SHIFT

Comandos avanzados DOS

Comando	I/E	Objeto	Formato
CTTY	I	Cambio por una consola auxiliar	CTTY nombre-unidad
EXE2BIN	E	Convierte los ficheros de formato ".EXE" en ficheros de formato ".COM"	EXE2BIN [d:][camino]nombre de fichero[.ext] [d:][camino][nombre de fichero[.ext]]
FIND	E	Busca las cadenas especificadas en los ficheros indicados	FIND [/V][/C][/N]cadena[d:] [[camino]nombre de fichero[.ext]...]
MORE	E	Visualiza los datos en pantallas enteras sucesivas	MORE
PROMPT	I	Establece un nuevo mensaje de solicitud	PROMPT[texto solicitud]
SET	I	Introduce las cadenas en el ambiente del procesador de los comandos	SET[nombre=[parámetro]]
SORT	E	Ordena los datos del texto	SORT[/R][/+n]

BIBLIOGRAFIA

Sistemas operativos.
Brinch Jansen. *Escuela de Informática de la Universidad Politécnica de Madrid.*

Sistemas operativos de la computación.
Davis, 1985. *Fondo Educativo Interamericano.*

Sistemas operativos.
Varios autores, 1985. *CEDED.*

Sistema operativo EP/M. Guía del usuario.
T. Hogan, 1983. *Ed. McGraw-Hill.*

Sistema operativo MS-DOS. Guía del usuario.
P. Hoffman / T. Nicoloff, 1985. *Ed. McGraw-Hill.*

Sistema operativo Unix. Guía del usuario.
R. Thomas / J. Yates, 1985. *Ed. McGraw-Hill.*

MS-DOS paso a paso.
Pinaud. *ELISA.*

CP/M palabra a palabra.
Dargery. *ELISA.*

NOTAS



emasiado a menudo encontramos personas que ante la sola idea de tener que utilizar un ordenador personal o cualquier otro instrumento informático se muestran aterrorizadas. Hay quien encuentra difícil hasta el simple accionamiento del mando a distancia del aparato de televisión. No obstante, las máquinas "inteligentes" están invadiendo todos los sectores: ya no debemos ofendernos si el cajero automático nos da la bienvenida, nos informa de los pasos que debemos dar para retirar dinero o nos felicita en nuestro cumpleaños, y tampoco ha de maravillarnos si podemos enviar por teléfono mensajes conversando con una gentil voz femenina que, en realidad, es generada por síntesis vocal por un ordenador.

En este volumen de la B.B.I. trataremos de explicar de modo claro y sencillo cómo es posible controlar un ordenador personal mediante el sistema operativo MS-DOS, uno de los más extendidos para ordenadores de 16 bits basados en el microprocesador 8086/8 de Intel, y adoptado como estándar por IBM para su famoso "PC". Proporcionamos toda la información que puede necesitar un principiante para tomar confianza con un ordenador personal sin poseer conocimientos específicos anteriores. Pero esta información es la misma que sirve también a quienes utilizan, de manera más o menos "transparente", paquetes de programas de aplicación y quieren conocer mejor las posibilidades del ordenador que tienen a su disposición.

Es un camino para introducirse en el mundo del MS-DOS con ejemplos prácticos e ilustraciones claras de casos particulares.